基于领域本体的游戏攻略文本标注算法研究与实现

陈小红[1] 陈环环[2] 方之家[2] 阮 彤[2] 王昊奋[2]

[1](盛大游戏 上海 201203)

[2](华东理工大学计算机科学与工程系 上海 200237)

摘 要 游戏门户网站为提升玩家们的游戏体验,建立了大量站点用以提供游戏资讯及相关攻略。然而这些站点间异构现象明显,且缺乏统一的知识体系。提出基于领域本体的文本标注算法,通过融合站点间的数据,构建游戏领域本体。同时,针对游戏领域的应用,优化了新词发现算法,并进一步对攻略文本进行语义标注。通过这些语义标签,不仅能直观地了解攻略中的内容,也能更好地为攻略文本的语义检索服务。实验证明,所提出的本体构建方法在游戏领域具有一定的推广性,同时游戏领域词汇发现算法与传统的分词工具相比也取得了更好的结果。

关键词 领域本体 游戏领域词汇发现算法 语义标注

中图分类号 TP391 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2017.02.014

RESEARCH AND IMPLEMENTATION OF ANNOTATION ALGORITHM FOR WALKTHROUGH TEXT BASED ON DOMAIN ONTOLOGY

Chen Xiaohong[1] Chen Huanhuan[2] Fang Zhijia[2] Ruan Tong[2] Wang Haofen[2]

[1](*Shengda Game Limited*, *Shanghai 201203*, *China*)

[2](*College of Information Science and Engineering*, *East China University of Science and Technology*, *Shanghai 200237*, *China*)

Abstract Nowadays, game web portals set up plenty of websites, providing game information and related walkthroughs, for players to enhance their gaming experience. However, these sites has obvious isomerism and lacks unified hierarchy. Thus, an annotation algorithm based on domain ontology is proposed. It is started with a data fusion step from a set of web portals to build a game domain ontology. Meanwhile, the neologism discovering algorithm is optimized according to its application in game domain, and the semantic annotation for walkthrough text is further developed. Thus these semantic tags not only embody the intent of each guides, but also serve the semantic search for walkthrough text. Experiments have proved that the proposed ontology construction method is scalable. Moreover, the optimized domain vocabulary discovering algorithm has a better result compared with the traditional segmentation tools.

Keywords Domain ontology Game domain vocabulary discovering algorithm Semantic annotation

0 引 言

随着游戏产业在我国迅速的发展,游戏的种类和数量也在不断地增加。为了能够让玩家对游戏有一个更为全面的认识,各大游戏资讯网站都开发了自己的游戏攻略主页。在这些站点中,不仅描述了各个游戏的基础知识,还有大量玩家自己撰写的游戏攻略。游戏玩家通过阅读这些游戏攻略,可以快速入门,并且掌握大量与游戏相关的进阶知识,可以有效地提升玩家的游戏体验。

然而,对于目前大多数的主流游戏而言,游戏攻略大多分散在各大资讯网站之中。当玩家需要查阅攻略时,通常需要辗转于多个资讯网站之间进行检索。同时,只有部分资讯网站提供了攻略搜索功能,玩家需要耗费大量时间去寻找与自己的游戏内容相关的攻略。

收稿日期:2015-12-11。国家自然科学基金项目(61402173);上海市经信委软件和集成电路产业专项资金(140304)。**陈小红**,高工,主研领域:大数据和虚拟世界的用户行为。**陈环环**,硕士生。**方之家**,硕士生。**阮彤**,副教授。**王昊奋**,博士。

这些问题不同程度降低了这些资讯网站的用户体验。为了能够解决这些问题,本文提出了一种基于领域本体的游戏攻略文本标注算法。使用本体建立起一套统一的描述方法,从而融合了各大资讯网站以及游戏内部的数据。基于该本体,可以有效地通过实体链接的思想,将攻略中的文本内容映射到本体中的相关概念,从而达到语义标注的目的。玩家可以通过这些语义标签,更快更全面地了解到攻略中所包含的信息,以便更快地定位到与自己相关的游戏内容;同时语义标签的生成,还可以为这些游戏站点构建更高效、更有实用价值的语义搜索系统,为玩家们提供更为精准的检索服务。

本文的主要贡献有以下几点:

(1) 针对资讯网站间的数据异构问题,提出了使用本体进行建模的思想。通过对游戏数据库、资讯网站导航页面以及玩家论坛中的数据进行融合,从而针对游戏内容本身构建了一个统一的本体。

(2) 在原有的基于大规模语料的领域词汇发现算法[1]的基础上,本文针对游戏术语简称、游戏内容用语等游戏领域词汇的用法及规则进行了优化,并生成了一系列可用于实体链接的锚文本。

(3) 基于实体链接[2]的思想,提出了对游戏攻略文本进行语义标注的算法。利用已构建好的游戏本体中的层次结构以及实例集合,对由锚文本所产生的候选歧义实体进行去歧义,选择最为契合的实体作为链接实体,从而生成语义标签。

1　相关工作

文本中的术语大多以简称的形式存在,因此需要构建游戏领域知识库存储术语全称,从而和文本建立一一映射关系。DBpedia、Freebase、YAGO 都是通用知识库,包含了丰富的数据,但是缺乏游戏领域知识。DBpedia[9]是从维基百科中自动抽取结构化信息,被广泛用于语义万维网和商业环境。Freebase[10]所有内容由用户添加,所有条目都采用结构化数据的格式。YAGO[11]主要信息来源于维基百科,具有足够高的准确度和覆盖度。本文所建的知识库和以上几个知识库相似,都是从互联网资源及百科资源进行数据融合得到。故本文针对游戏领域,从三个不同数据源爬取了游戏领域相关知识,构建了游戏领域知识库。

文本标注技术是信息抽取的一个应用,在过去的几年中得到了广泛的研究。Mihalcea 等[3]提出了排序算法 TextRank,主要思想是将文本看成一个词的网络,网络中的链接表示词与词之间的语义关系,但是该方法不适用于稀疏文本。Park 等[4]等提出了基于 sigmoid 贝叶斯模型的关键词自动抽取方法,解决了数据稀疏的问题,然而该方法要求数据值必须遵循 sigmoid 分布才能在贝叶斯结构中表现出来,具有局限性。本文面向的是游戏领域,游戏攻略内容表述比较偏口语化,直接提取关键词或主题词比较困难,且攻略标题包含大量游戏术语,因此提出了通过提取文本数据中游戏术语作为文本标签的思想。

术语抽取是本文攻略文本标注的关键技术,现有的领域术语研究主要分为:基于规则的方法、基于统计的方法和规则与统计相结合的方法。基于规则的方法主要利用术语词典和规则模板来进行术语抽取。Buitelaar等[5]提出利用不同词性的组合规则得到名词性词组,然后利用过滤算法得到领域术语。基于规则的方法比较简单,但是要求规则编写人员具有丰富的语言知识。基于统计的方法是利用术语内部各组成成分之间较高的关联程度及术语的领域特征信息来抽取术语。Tomokiyo 等[6]提出利用语言模型之间的相对熵来计算词之间的耦合度和术语的领域相关性。基于统计的方法[7]不局限于一种领域,通用性较强,但是算法性能依赖于语料库规模的大小和候选术语的词频,将一些低频率但合法的术语忽略掉,不适用于稀疏的文本。Sui 等[8]提出使用统计方法计算术语的置信度,然后使用规则过滤领域术语候选,这是将统计和规则相结合的方法,但是术语抽取的准确率还是没有达到理想水平。本文使用游戏领域词汇发现算法对术语进行抽取,它不依赖任何词库,针对稀疏的文本也能达到很好的效果。

2　整体思想

2.1　问题定义

游戏的资讯网站包含了丰富的游戏内容,为玩家了解游戏提供了方便的平台。资讯网站数据库包含了大量的游戏术语,是构建游戏领域本体的主要来源,具有较高的可靠性及通用性。资讯网站中导航页帮助玩家快速找到想要的游戏内容,具有一定的分类结构,作为领域本体构建的补充。游戏论坛中玩家总结了大量的游戏术语,这些术语经常出现在攻略数据中,进一步丰富了领域本体。本文是从以上三个数据源构建游戏领域本体存储到知识库中,具体构建方法在第 3 节中会详细介绍。本文所用语料来源于资讯网站中游戏攻略文本数据,每一条数据包含“标题”和“内容”两部分,大多标题中都会出现游戏术语或其简称。本文的

输入为大量的游戏攻略文本数据,输出为所有数据的标签。本文的主要任务可以描述为:对每条数据,抽取出标题中所包含的游戏术语或其简称的集合 $M(m_1, m_2, \cdots, m_n)$,在知识库 $E\{e_1, e_2, \cdots, e_m\}$ 中找到集合 M 中每个元素所代表的游戏术语或其简称所对应的游戏术语全称,即集合 $A\{a_1, a_2, \cdots, a_n \mid a_1, a_2, \cdots, a_n \in E\}$。

以图 1 给出的攻略数据为例,标题中出现了“黑魔”,将“黑魔”和知识库进行链接,“黑魔”在知识库中所对应的全称为“黑魔法师”,“黑魔法师”即是该数据的标签。在构建知识库时,将爬取的资讯网站和游戏论坛数据转化成标准的 Ontology[12] 语言格式,在知识库中存储成如图 1 所示的结构(只给出了部分结构),共包含两层:模式(schema)层和实例(instance)层。模式层表达了类与类之间的关系,即父类与子类的关系。如“攻略”是顶层类,“攻略”下包含“职业”和“副本”等子类,职业类下又包含“基础职业”和“进阶职业”等子类。知识库结构的最底层是实例层,如进阶职业类下包含“黑魔法师”等实例。

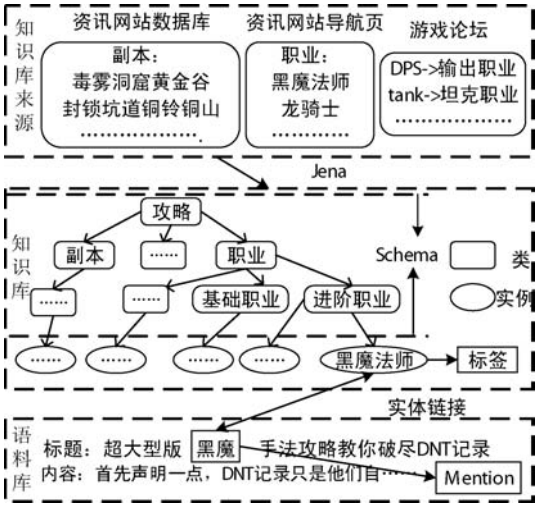

图 1 基于领域本体的文本标注

2.2 整体流程

基于领域本体的游戏攻略文本标注主要包括三个阶段(如图 2 所示):构建知识库、抽取领域词汇、实体链接。本文知识库的来源多样化,从三个不同的数据源爬取游戏领域知识构建游戏领域本体存储在知识库中,知识库中存在的游戏术语全称称为“实体”。然后从多个资讯网站中抽取游戏攻略文本数据作为语料库,每条攻略数据包括“标题”和“内容”两部分内容。使用游戏领域词汇发现算法从语料中抽取出游戏领域词汇称之为“指代项”(Mention)即可进行实体链接的锚文本,其包含游戏术语全称或简称和其他游戏词汇。对每一个标题判断是否包含“Mention”,若包含“Mention”,则和知识库中的实体使用匹配算法进行链接,链接到的游戏术语的全称即文本标签从而对攻略数据进行语义标注。

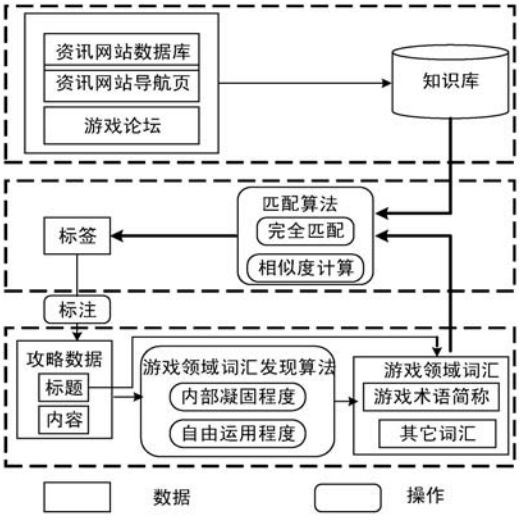

图 2 整体流程

3 构建知识库

本节将分别从知识库结构的模式层与实例层,详细阐述如何融合多个资讯网站的游戏知识,并同时对融合过程中所产生的冲突予以解决。首先,在构造知识库模式层的过程中,本文利用两部分的数据源,分别是资讯网站提供的导航栏,以及网站以网页方式提供的游戏数据库。通过解析网页的 HTML 结构,可以得到该导航栏或知识库定义的分类目录结构。例如:“装备”在资讯网站数据库分类目录下有“头部防具”、“身体防具”、“手部防具”、“腰部防具”等。故在知识库的结构中,“头部防具”、“身体防具”等均可作为“装备”的子类。

本文将位于各资讯网站分类目录中叶节点的数据作为知识库结构的实例层。例如,上述例子中,“头部防具”类下还包含有叶节点“风化兜帽”、“风化眼睛”等,均存在唯一的 URI 可以链接到具体介绍“风化兜帽”、“风化眼睛”等的页面。因此,“风化兜帽、风化眼睛”将作为“头部防具”的实例存储于知识库中。

此外,游戏论坛中会总结游戏攻略中玩家自定义的游戏术语和资讯网站游戏术语形成的同义词表。本文对游戏论坛进行抽取,并将同义词表存储于知识库中。例如,游戏攻略中玩家通常会用英文的简写来表征游戏术语,使用“DPS”来表示“输出职业”,因此,“DPS”作为“输出职业”的同义词存储于知识库中。

在融合资讯网站构建知识库的过程中,由于网站间的异构问题,会产生两种冲突。其一是在融合模式

层时,同一个类在不同网站所属父类不同。其二是在融合实例时,同一个实例在不同网站的类别不同。

融合网站间模式出现的冲突,可定义为对于类 M,在不同的网站中分别归属于两个不同的父类 E 和 C。本文将通过判断类 E 和类 C 之间的关系来解决冲突。计算公式为:

$$I_{E_i,C_j} = \frac{n_{E_i} \cap n_{C_j}}{n_{E_i} \cup n_{C_j}} \tag{1}$$

其中,n_{e_i} 表示在 i 网站 E 类中实例数目,n_{c_j} 表示 j 网站 C 类的实例数目。I_{E_i,C_j} 代表这两个类的实例重叠比。若该值较高,则本文将类 E 和类 C 视为相等关系;反之,则可认为类 E 和类 C 之间是父类和子类关系。对于后者,将进一步复用式(1)计算类 M 与类 E 的实例重合比 $I_{M,E}$ 以及类 M 与类 C 的实例重合比 $I_{M,C}$。若 $I_{M,E} > I_{M,C}$,则认为 E 是 C 的父类,否则 C 是 E 的父类。例如,在“多玩(http://ff14.duowan.com/index.html)”网站中“剑”是“单手武器”的一个子类,但是在“178(http://ff14.178.com/)”网站中“剑”是“武器”的一个子类,因此,融合时会产生冲突。通过式(1)计算可知,“武器”是“单手武器”父类,最后形成的关系为“单手武器”是“武器”的子类,“剑”是“单手武器”的子类。

在处理实例融合产生冲突时,我们会综合考虑每个实例所在类的出现频次以及该类的丰富程度,进而形成一个综合指标来解决冲突。每个实例所在类的出现频次可通过式(2)进行归一化计算:

$$F_i = \frac{M_i}{N} \tag{2}$$

其中 N 为待融合的资讯网站,M_i 为该冲突实例属于 i 类的网站数。本文共融合 4 个资讯网站。例如,有 3 个网站将实例“采矿工”归入“采集职业”类,而仅有 1 个网站将其归入“大地使者”类,根据式(2)可别得出归一化值为$\frac{3}{4}$和$\frac{1}{4}$。当该指标越高时,说明有更多网站倾向于将实例划分在该类下。另一方面,本文利用如下公式计算每个类的丰富程度:

$$R_i = \frac{S_i}{S_1 \cup S_2 \cdots \cup S_i} \tag{3}$$

其中,S_i 为 i 类的实例数目。“采集职业”类的实例数目为 12,“大地使者”类的实例数目为 7,两个类的实例并集为 15,故分类完整度分别为$\frac{12}{15}$和$\frac{7}{15}$。由式(2)和式(3)可得到综合指标公式:

$$\max\{F_i \times R_i\} \tag{4}$$

对于上述例子,可根据式(4),计算得到“采矿工”作为“采集职业”类的综合指标值较大,因此,最终“采矿工”作为“采集职业”的实例存储于知识库中。

4　实体链接

4.1　游戏领域词汇发现

对于中文来说[7],文本中没有空格标志词语边界,没有首字母大写等明显特征来表征一个术语。术语识别过程通常要和中文分词过程相结合,然而汉语词汇具有开放性,无论建立多大的词典,都不可能包含所有的词汇,而且随着时间的推移还会不断出现大量新词。使用传统的分词方法处理起来比较困难且难以达到预期的效果,在此基础之上,我们提出了游戏领域词汇发现算法。

游戏领域词汇发现算法是在新词发现算法基础之上所做的改进。新词发现算法[1]是顾森提出来的一种基于大规模语料挖掘新词的方法。他所使用的语料是人人网 2011 年 12 月前半个月部分用户状态,而本文所使用的语料是游戏攻略数据。由于语料库有所不同,因此在此基础上进行了改进得到游戏领域词汇发现算法。游戏领域词汇发现算法有两个成词标准分别是“内部凝固程度”和“自由运用程度”。凝合度的计算公式为:

$$C(ABC) = \frac{P(ABC)}{\max\{P(AB)P(C), P(A)P(BC)\}} \tag{5}$$

$$P(X) = \frac{N(X)}{length(text) - length(X) + 1} \tag{6}$$

其中 N 为出现次数,P 为出现的概率,$length$ 为所含字数。在式(5)中“ABC”表示要抽取的词由 A、B、C 三部分组成,实际操作时,候选词可能不止这三部分,其计算公式相同。文本片段的自由运用程度也是判断其是否成词的标准之一,在此,用“信息熵”衡量。具体计算公式如下所示:

设 Z 的左邻字集合为$\{X_1, X_2, \cdots, X_n\}$,则 Z 的左邻字信息熵为:

$$H(X) = \sum_{i=1}^{n}(-P(X_i)\log P(X_i)) + \sum_{i=1}^{m}(-P(LPunc_i)\log P(LPunc_i)) \tag{7}$$

其中 $n + m = N(Z)$,对上式化简后如下所示:

$$H(X) = \sum_{i=1}^{n}(-P(X_i)\log P(X_i)) +$$

$$\frac{N(LPunc)\cdot\log N(Z)}{N(Z)} \tag{8}$$

其中 $P(X)=\frac{N(XZ)}{N(Z)}$，N 为出现次数，$N(LPunc)$ 为左邻字是标点符号或开头的次数。

同理，设 Z 的右邻字集合为 $\{Y_1,Y_2,\cdots,Y_n\}$，化简后则 Z 的右邻字信息熵：

$$H(Y)=\sum_{i=1}^{n=}(-P(Y_i)\log P(Y_i))+\frac{N(LPunc)\cdot\log N(Z)}{N(Z)} \tag{9}$$

其中 $P(Y)=\frac{N(YZ)}{N(Z)}$，N 为出现次数，$N(LPunc)$ 为右邻字是标点符号或结尾的次数。

信息熵为 $\min\{H(X),H(Y)\}$。

本文提出的游戏领域词汇发现算法在原有新词发现算法基础之上做出以下改进：

（1）鉴于游戏中很多副本的名称往往很长，比如“山中战线泽梅尔要塞”、“剑斗领域日影地修炼所”。若所抽取候选词的长度太小，容易将较长的游戏术语分割成更小的词语组成；若所抽取的候选词的长度太大，所抽取的游戏术语中易混杂无关词组。经过反复实验，候选词语的长度设置为 10 效果最佳。

（2）在计算信息熵时，对于左邻字是标点符号或没有左邻字（词语本身是开头），以及右邻字是标点符号或没有右邻字（词语本身是结尾）的情况，视其左邻字或右邻字是一个全新的从未出现过的字，并在实际操作时分别统计该种左邻字与右邻字的出现次数，以做等效计算（见上述信息熵的计算式）。开头、结尾和标点符号的存在，起到了分词的作用，对于抽取领域词汇大有裨益。之所以不据此将词语的左邻字或右邻字信息熵直接设为无穷大，而仅仅是把每一个开头、结尾或标点符号视作一个全新的从未出现过的字，是为了提高新词发现的容错程度，以避免因为攻略中误出现标点符号或误作开头、结尾，而执行了错误的分词。使用此方法之后，如果攻略中偶尔出现一个笔误，则其对信息熵的计算不会造成太大的影响；如果是正常情况下多次出现标点符号或多次作为开头、结尾，由此便能通过提高信息熵而起到分词的作用。

（3）针对英文单词与纯数字的情况，我们在抽取候选词时，先将文本中所有出现的英文单词抽取加入到候选词集合中，之后再做正常抽取时，如果碰到词语是全英文或全数字，则不放入候选集，从而大大提高了抽取的准确率与效率。因为在中文汉字占主体的攻略文本中，偶尔出现几个连续的英文字母，其必然是一个完整的英文单词，英汉字符的不同已经能够很好地起到分词的作用。而如果将英文单词也放到正常的流程中抽取，则在抽取的过程中会将英文单词的子串也一并加入候选词集合，从而影响分词效率。至于不将全数字字符串加入候选集，是因为目的是抽取游戏术语，而纯数字一般不作为游戏术语出现，将其加入会增加干扰、影响抽取质量。

游戏领域词汇发现算法伪代码如下：

算法 1 游戏领域词汇发现算法

```
Input:setStrategies, Set of game strategy
Output:setPhrases, Set of phrase that meet conditions
1: for each strategysetStrategies do
2:   PhraseMap lengthd(phrase; num)
3:     if leftPhrase exists then
4:         LeftPhraseMap leftPhrase
5:     if leftPhrase is punctuation or beginning then
6:         PBMap (LeftPhrase, 1)
7:     rightPhrase same as leftPhrase
8:  if ! leftPhrase || ! rightPhrase then As a new word
9:  PhraseSet 1 < length < d(phrase)
10:     setPhrases(Phrase, C > 5&&H > 0.6&&Num4)
11: end for
```

最后凝合度、信息熵、出现的次数的阈值的选择是通过多次实验得到的最佳值。使用此算法后，得到比如“巴哈姆特大迷宫”、“极水神”、“炼金术师”等可用于实体链接的锚文本即 Mention 列表。

4.2 语义标注

游戏术语在文本中经常以简称的形式存在，例如“白魔法师”在文本中常常记为“白魔”，由于知识库中存储游戏术语的全称，因此需要进行链接。从而得到知识库中所对应的游戏术语全称即文本标签对攻略数据进行语义标注。文本中的简称无法直接和知识库中实体进行关联，因此需采取匹配算法。通过观察数据知每个标题包含的 Mention 个数可能不止一个，这种对应的标签即为多标签，如表 1 所示。Mention 可能和知识库中的实体完全匹配，或者 Mention 是知识库中实体的简称或缩写，或者 Mention 是知识库中实体相邻词组成的子串，或者 Mention 是知识库中实体不相邻字组成的子串（此时通过计算相似度进行匹配），或者知识库中实体是 Mention 的相邻字组成的子串等，如表 2 所示。基于以上观察结果，提出了进行链接的匹配算法，具体算法及思想在下面的内容中会详细介绍。

表 1　标题中的标签类型

标题	Mention	类别
手柄技能按键三页法排 布龙骑士武僧骑士键位	龙骑士、 武僧、骑士	多标签
《最终幻想 14》20 级最 快骑上陆行鸟的方法	陆行鸟	单标签

表 2　Mention 和实体之间的关系

Mention	实体	关系
龙骑士	龙骑士	相同
黑魔	黑魔法师	Mention 是实体的 相邻字组成的子串
巴哈大迷宫	巴哈姆特 大迷宫	Mention 是实体的 不相邻字组成的子串
摩杜纳服	摩杜纳	实体是 Mention 的子串

通过以上对数据的观察和分析,得出以下匹配算法:

算法 2　匹配算法

Input:

1) setMentions: Set of mentions in corpus;

2) setEntities: Set of entities in knoeledge base;

3) setStrategies: Set of strtegies, each of the strategy contains a title and the text;

Output: {(strategy, setWords)} pair list;

Initiation: The English letters in setMentions and setEntities and setStrategies are unified in Uppercase, remove the blank space in setStrategies;

```
1: if titile. contains(mention) then
2:   if mention. equal(entity) then
3:     setWordsentity
4:   else simlarRate(mention, entity)
5:         setWordslargestRate(entity)
6:     or else setWords is empty then
7: entity = entity. replace(mention, "") , setWordslength(entity)
8:     title = title. replace(mention, "")
9:   perform steps 1 - 6
10: mapStrategySetEntities(strategy, setWords)
```

在上述匹配算法中,相似度的计算公式为 $\frac{lcs.length \cdot lcs.length}{mention.length \cdot entity,length}$,setCurWords 是一条攻略中一个 mention 的所有对应 entity 集合, setWords 是一条攻略的所有对应 entity 集合即得到的每条攻略的标签集合。使用匹配算法将每条数据标题中所出现的游戏术语的全称或简称与知识库进行了映射,得到其对应的游戏术语全称即文本标签,使用这些标签对文本数据进行语义标注。

5　实验结果和分析

5.1　实验数据与设置

本文所用的语料库是“最终幻想 14”这款游戏不同资讯网站中每天都不断更新的游戏攻略文本数据,每条数据包括标题和内容两部分,爬取这些攻略文本数据并进行去重后,共计 1788 条数据。

虽然本文提出的方法是无监督的,但是还是需要人工对数据进行标注来评估标注质量。本文使用准确率 Precision 和召回率 Recall 作为评价标准,若 A 表示正确标注的数据条数,B 表示错误标注的数据条数,C 表示没有进行标注的数据条数,则准确率和召回率的定义分别如下:

$$Precision = \frac{A}{A+B} \qquad Recall = \frac{A}{A+C}$$

5.2　结果分析

在抽取文本游戏领域术语或其简称时,分别使用了游戏领域词汇发现算法、Ansj、N-grams(n 一般取 2 或 3,若 n 过大,会导致计算复杂度增高,且结果数据过于稀疏),最后分别统计每一种方法所得标注数据的准确率和召回率,如图 3 所示。错误标注的情况分别如下所示:

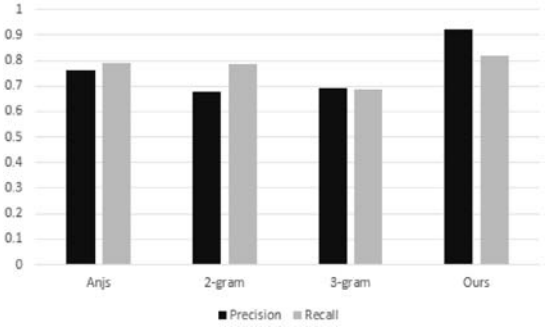

图 3　不同词汇抽取方法的比较

(1) 少标签。如果一个标题中出现的游戏术语或其简称不止一个,则这条文本数据属于多标签,在标注时可能会发生遗漏标签的情况。比如标题“FF14 龙骑和武僧输出大对比 谁更难操作?”,标题中出现的游戏术语或其简称为“龙骑”、“武僧”,该数据的正确标签为“龙骑士”、“武僧”。若最后给出的标签少了其中任何一个,都认为标注错误。

(2) 多标签。比如标题“暴击流黑暗魔法师在2.4版本中装备怎么进阶”,其正确的标签为:“2.4 版本、

黑魔法师、装备”。若最后给出的标签为“2.4 版本、黑魔法师、装备、白魔法师”,相对于正确的标签多出了“白魔法师”标签,则认为该标注是错误的。

(3) 标签错误。比如标题“最终幻想 14 数据分析告诉你骑士起手仇恨究竟有多高”,正确的标签为“骑士”。若最后给出的标签为“龙骑士”则认为该标注是错误的。

对于没有标注的数据,据观察文本内容多偏向于玩家心情类的攻略,与游戏本身并不相关。比如标题为“FF14 让你更快适应游戏之你所不知道的小窍门集合”,标题中没有出现与游戏术语相关的词汇,因而本文所提出的方法无法进行标注。

从图 3 中可以看出,使用游戏领域发现算法所得标注的准确率和召回率明显高于 Ansj、2-gram、3-gram。Ansj 是基于词典的分词方法,针对新闻、文章等普通文本的分词,能够得到比较高的准确率,而本文语料数据包含的游戏术语多,使用 Ansj 大多将游戏术语分成更细粒度的词,比如“白魔法师”会分成“白魔”和“法师”两个词,因此标注效果难以满足要求。N-grams 需要相当规模的语料来确定模型参数,N 不宜取过大,N 值一般为 2 或 3。有些游戏术语由 3 个或 4 个字组成,使用此语言模型会将这些词分开,因此和知识库进行链接时匹配率不高,会生成与文本数据无关的标签。

6 结 语

本文主要研究了游戏领域中攻略文本的标注算法,在对同一个游戏的不同资讯网站进行信息融合的基础之上,建立了统一的本体;提出了一种全新的领域词汇抽取算法即游戏领域词汇发现算法,并据此从语料库中抽取出文本标题所包含的游戏术语或其简称,再和知识库进行链接,最终得到文本数据的标签。在抽取游戏领域词汇时,将游戏领域词汇发现算法与 Ansj 和 N-grams 这两个已有的自然语言处理工具进行对比,结果显示该方法在领域词汇抽取方面具有优越性。本文的优点是将文本标注问题看成是实体链接问题,并且解决了在领域词汇抽取工作上一直存在的抽取质量问题。数据集上的结果表明,本文提出的方法达到较高的准确率和召回率。接下来计划将攻略标签用于游戏攻略上的语义搜索和将游戏领域词汇抽取方法应用于其他领域进行领域词汇抽取,如计算机领域、医疗领域等。

参 考 文 献

[1] 顾森. 基于大规模语料的新词发现算法[J]. 程序员,2012(7):54-57.

[2] Meij E, Balog K, Odijk D. Entity linking and retrieval[C]// Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2013:1127.

[3] Mihalcea R, Tarau P. TextRank: bringing order into texts [C]//Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004:401-411.

[4] Park Y C, Han Y S, Choi K S. Automatic thesaurus construction using Bayesian networks[C]//Proceedings of the Fourth International Conference on Information and Knowledge Management. ACM, 1995:212-217.

[5] Buitelaar P, Olejnik D, Sintek M. A protégé plug-in for ontology extraction from text based on linguistic analysis[C]// First European Semantic Web Symposium. Springer, 2004:31-44.

[6] Tomokiyo T, Hurst M. A language model approach to keyphrase extraction[C]//Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment. Association for Computational Linguistics, 2003: 33-40.

[7] 季培培,鄢小燕,岑咏华. 面向领域中文文本信息处理的术语识别与抽取研究综述[J]. 图书情报工作,2010,54(16):124-129.

[8] Sui Z, Chen Y, Hu J, et al. The research on the automatic term extraction in the domain of information science and technology[C]//Proceedings of the 5th East Asia Forum of the Terminology, Haikou, Hainan, China, 2002:444-451.

[9] Lehmann J, Isele R, Jakob M, et al. DBpedia-a large-scale, multilingual knowledge base extracted from Wikipedia[J]. Semantic Web, 2015, 6(2):167-195.

[10] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. ACM, 2008:1247-1250.

[11] Suchanek F M, Kasneci G, Weikum G. Yago: a core of semantic knowledge[C]//Proceedings of the 16th International Conference on World Wide Web. ACM, 2007:697-706.

[12] Aref M M, Zhou Z. The ontology web language (OWL) for a multi-agent understating system[C]//Proceedings of the 2005 International Conference on Integration of Knowledge Intensive Multi-Agent Systems. IEEE Computer Society, 2005:586-591.