基于改进的 Lesk 算法的词义排歧算法*

王永生
(同济大学 出国培训学院,上海 200092)

摘 要:英文中的一词多义现象非常普遍,这给英文的词义排歧带来了极大的困难。针对这种情况,提出了一种基于改进的 Lesk 算法的词义排歧算法,即以语义词典 WordNet 为基础,借助 CBC 算法扩充目标词的相似词集合,通过改进的 Lesk 算法进行词义排歧。算法以英文 Senseval-2 任务作为测试目标,通过对目标词的义项进行筛选,去除其中一些不常用的义项,实验结果表明,总体排歧正确率达到 58.4%。

关键词:词义排歧;Lesk 算法;WordNet
中图分类号:TP391 **文献标识码**:A **文章编号**:1674-7720(2013)24-0069-03

Word sense disambiguation using an adapted Lesk algorithm

Wang Yongsheng
(International Education College, Tongji University, Shanghai 200092, China)

Abstract: In English, lexical ambiguity is pervasive, so English word sense disambiguation is one of the most difficult tasks in natural language processing. This paper presents an adaptation of Lesk algorithm based on WordNet. Additionally an algorithm called CBC is used to enlarge the amount of similar words of the target word. This method is evaluated using the Senseval-2 word sense disambiguation exercise, and attains an overall accuracy of 58.4%.

Key words: word sense disambiguation; Lesk algorithm; WordNet

词义歧义(Ambiguity of Word Senses)是自然语言的基本特征。以语义词典 WordNet 为例,其所有动词中多义词的平均义项数为 3.57 个[1],这就给英语的词义排歧 WSD(Word Sense Disambiguation)带来了极大的困难。WSD 的研究伴随着 20 世纪 40 年代机器翻译的兴起而产生,经过几十年的发展,已取得相当大的进展。概括来讲,WSD 主要有 3 种方法[2]:

(1)基于词典的方法(Dictionary-based Methods)

基于词典的方法也称基于知识的方法(Knowledge-based Methods),这种方法使用机器可读词典(Machine Readable Dictionary)给词义排歧提供义项(Senses)以及相应义项的上下文(Context)。这类方法比较典型的有 Lesk 算法[3]及选择限制法(Selectional Restrictions)[4]等。

(2)无指导的方法(Unsupervised Methods)[5]

无指导的方法使用未进行人工义项标注(Sense tagging)的语料库,通过机器学习方法来进行词义排歧。

(3)有指导的方法(Supervised Methods)[6]或半指导方法(Semi-supervised Methods)[7]

与无指导的方法相对应,有指导或半指导方法主要使用人工义项标注语料库进行机器学习。

在这 3 种方法中,基于标注语料库的指导方法的排歧效果最好[2]。然而这种方法需要大规模的义项标注语料库。目前现有的此类语料库基本都是试验性的、小规模的,如果自己动手创建,显然也不是短期所能完成的。与此同时,这类方法也受限于标注语料库的大小,难以做到全词(All-words)排歧[2]。

本文采用一种改进的 Lesk 算法来进行词义排歧。Lesk 算法通过查找机读词典,将待排歧词(下文统一称为目标词)的每个义项的定义与上下文中词的定义进行匹配,单词重叠最多的义项胜出[3]。

分布假设理论(Distributional Hypothesis)认为出现在相同上下文中的词往往是相似的[8]。基于这一点,本文从扩大目标词的相似词(Similar Words)的规模入手来对算法进行改进。主要从两方面来入手:

*基金项目:教育部人文社会科学研究基金青年项目(07JC740009)

(1)采用改进的 CBC 算法(Clustering By Committee)[9]来从 WordNet 外部扩充目标词每个义项对应的相似词集。

(2)除了 WordNet 中的上义关系,还采集 WordNet 中的其他一些关系(如下义关系等)来扩充相似词的规模。以 plant 的“工厂”义项为例,construction 是它的一个上义关系,与它的相似度极高,而它的下义关系“manufactory”、“mint”显然也是相似度极高的词。

1 基于 WordNet 的改进的 Lesk 词义排歧算法

1.1 获取目标词每个义项对应的相似词

要获取目标词的相似词,首先来定义用于测量两个词相似度的公式(Similarity Measure)。概括地讲,词汇相似度的计算方法主要有两类,一类基于分布假设理论(Distributional Hypothesis),该理论基于以下假设,即出现在相同上下文中的词往往是相似的[8];另一类基于点互信息 PMI(Pointwise Mutual Information),PMI 表示两个词之间的相关性,PMI 值越大,两个词之间的相关性就越大,其相似程度也就越高。研究表明,PMI 作为词汇相似度计算标准要优于分布假设理论[10]。

CBC 算法通过定义一个特征向量(Feature Vector)来定义目标词,而这个特征对应着该词所在的上下文,特征的值就是特征与目标词的点互信息。

要获取目标词的相似词,首先定义两个词 w1 和 w2 之间的相似度计算公式:

$$\mathrm{PMI}(w1,w2)\approx\ln\left(\frac{f_d(w1,w2)N}{f_{w1},f_{w2}}\right)$$

其中,f_{w1} 和 f_{w2} 分别表示单词 w1 和 w2 在语料库中出现的次数;$f_d(w1,w2)$ 表示单词 w1 和 w2 在语料库中长度为 d 个词的窗口内共同出现的次数;N 表示语料库中所有词的个数。

很显然,PMI 定义了词 w1 和 w2 之间的相关性(Correlation),该值越大,说明这两个词共同出现的概率就越高。

但由于 PMI 众所周知的偏置问题(Biased Problem),即它往往过于强调不常用词的关联度[11],因而,Han Lushan 等人通过修正 PMI 公式,将之重新定义成 PMI_{max}[10]:

$$\mathrm{PMI}_{max}(w1,w2)=\ln\left(\frac{(f_d(w1,w2)-x)N}{\frac{f_{w1}f_{w2}}{s_{w1}s_{w2}}}\right)$$

其中 s_{w1} 和 s_{w2} 分别表示词 w1 和 w2 的义项个数;x 为偏移系数,定义如下:

$$x=\frac{e^k}{N}\left(f_{w1}f_{w2}-\frac{f_{w1}f_{w2}}{s_{w1}s_{w2}}\right)$$

经过实验,主要的几类词对应的系数 k 的值如表 1 所示。

表 1 不同类型的词对应的系数 k 的值

	e^k	k
名词	30	3.4
动词	40	3.7
形容词	70	4.2
副词	40	3.7

本文通过上述 PMI_{max} 工具,基于 BNC 语料库就可以取得目标词的相似词集。然后通过 CBC 算法(即一个 3 步的聚类算法)进行聚类,从而得到某个目标词的所有义项对应的相似词集[9]。以名词 plant 为例,依据 CBC 算法,其在 WordNet 中的 4 个义项对应的相似词集如下:

义项 1:factory, facility, mill, works, refinery

义项 2:shrub, perennial, bulb, flora

义项 3:trap

义项 4:actor, performer

1.2 选取目标词的其他一些关系

一味地选取上义词,其成员太少,相似度差,而通过查看 WordNet 的其他关系可以发现,其中许多词出现在通过 CBC 算法得出的相似词的集合中。以名词 plant 的“植物”义项为例,通过 WordNet TreeWalk 展示出的第 1 个义项的下义词有 11 个,而第 2 个义项展示出的下义词有 32 个之多,非常可观。因而不限于上义关系,将目标词的其他一些关系也纳入考虑范围。

1.3 改进的 Lesk 词义排歧算法

算法描述如下:

首先定义一段文字 S:

$S=(W_{-n},\cdots,W_{-1},W_0,W_1,\cdots,W_m)$

其中,W_0 为要进行语义排歧的多义词,即目标词;$W_i(i=-n,\cdots,-1,1,\cdots,m)$ 组成了目标词 W_0 的上下文,为名词、动词、形容词或副词,S 中除此以外的词被过滤。

假设 W_0 在 WordNet 中有 N 个义项,则定义:

$S_i=\{(\mathrm{Word}_s,\mathrm{Weight}_s)\},i=1,\cdots,N$

其中 Word_s 和 Weight_s 分别表示目标词 W_0 的第 i 个义项对应的语义词(或相似词)及其权值,它们组成一个二元组,并组成集合 S_i。

再定义:

$C=\{(\mathrm{Word}_c,\mathrm{Weight}_c)\}$

其中 Word_c 和 Weight_c 分别表示从目标词的上下文中的词在 WordNet 中的所有义项的某个关系的层次结构中提取的语义词(或相似词)及其权值,它们也是一个二元组,组成集合 C。

再定义:

Sense_i 为目标词 W_0 的第 $i(i=1,\cdots,N)$个义项。

Score_i 为目标词 W_0 的第 $i(i=1,\cdots,N)$个义项的得分,其初始值均为 0。

则目标词 W_0 的第 $i(i=1,\cdots,N)$个义项的得分为:

$\forall(\mathrm{Word}_s,\mathrm{Weight}_s)\in S_i,\ \forall(\mathrm{Word}_c,\mathrm{Weight}_c)\in C$

IF $\mathrm{Word}_s=\mathrm{Word}_c$ THEN $\mathrm{Score}_i=\mathrm{Score}_i+\mathrm{Weight}_s\times\mathrm{Weight}_c$

则最终在文本 S 中的多义词 W_0 的词义为:

$$\mathrm{Sense}_{Max}=\underset{\mathrm{Sense}_i}{\mathrm{argmax}}\ \mathrm{Score}_i,i=1,\cdots,N$$

也就是说,如果某个 $\mathrm{Score}_i(i=1,\cdots,N)$ 在 N 个义项中得分最高,则在文字 S 中目标词 W_0 的词义就是第 i 个义项 Sense_i。

算法的思想其实很简单,就是将 S_i 集合分别与 C 集合中的语义词进行比较,如果其中的某两个语义词相

同，则将两者的权值相乘作为这次匹配的得分，并累加作为该集合的得分，最终哪一个集合的得分高，则对应的义项胜出。整个算法如图1所示。

2 算法评估

为了对上述算法进行评估，本文首先采用英文Senseval-2任务中的29个名词作为测试目标，其测试情况如下：

art(58.2%, 4, 98), authority(47.8%, 7, 92),
bar(19.2%, 17, 151), bum(60.0%, 6, 45),
chair(60.9%, 6, 69), channel(15.1%, 10, 73),
child(70.3%, 4, 64), church(57.8%, 4, 64),
circuit(41.2%, 7, 85), day(22.8%, 10, 145),
detention(75.0%, 2, 32), dyke(64.3%, 3, 28),
facility(43.1%, 5, 58), fatigue(39.5%, 6, 43),
feeling(41.2%, 6, 51), grip(15.7%, 11, 51),
hearth(81.3%, 3, 32), holiday(80.6%, 3, 31),
lady(69.8%, 3, 53), material(37.7%, 11, 69),
mouth(40.0%, 11, 60), nation(75.7%, 4, 37),
nature(56.5%, 5, 46), post(26.6%, 20, 79),
restraint(48.9%, 6, 45), sense(50.9%, 6, 53),
spade(57.6%, 4, 33), stress(41.0%, 8, 39),
yew(82.1%, 2, 28)

其中每个词后面的括号内有3个数字，分别表示该词的词义排歧正确率、在WordNet中的义项数、该词的测试例句数。在这29个名词中，平均每个词有6.69个义项。

经过测试，全部29个名词的1 754个样例中只有795个正确，整体正确率只有45.2%。显然这个结果很不理想，有没有进一步改进的余地呢？事实上，以名词plant为例，它在WordNet中共有4个义项，分别表示“工厂”、“植物”、“内线，卧底”和“假装观众的演员”，很明显，只有前两个义项最常用，其中第4个义项甚至只有美国俚语中才会出现。也就是说，如果去除这两个不常见的义项，毫无疑问会极大地提高排歧的正确率。以BNC语料库和SemCor语料库为参照，这29个名词的义项只有出现在这两个语料库中才被选取，这样平均义项就由6.69下降到2.87。然后再使用本文的排歧算法进行排歧，其整个排歧正确率可大幅提高到58.1%。如果扩充到Senseval-2任务的全部402个测试词，整体的排歧正确率为58.4%。虽然此结果与排歧效果最好的监督学习方法(约64%的正确率)相比还有一定差距，但与传统的Lesk算法在Senseval-2任务中取得的51.2%的排歧正确率相比，已有了较大的提高[2]。

本文以语义词典WordNet为基础，借助改进的CBC算法扩充目标词的相似词集合，并通过改进的Lesk算法进行英语词义排歧。算法首先以英文Senseval-2任务中的29个名词作为测试目标，起初排歧正确率只有45.2%，很不理想，但经过对目标词的义项进行筛选，去除其中一些不常用的义项，再进行测试，则整个排歧正确率可提高到58.1%。而如果扩充到全部402个测试词，整体的排歧正确率为58.4%。

但由于算法运行时需大量搜索WordNet的语义数据库，从而导致算法运行速度较慢。虽然可以通过事先创建多义词的语义词库来提高速度，但无论如何，该算法的运行速度都会比基于统计学的算法要慢，这也是该算法的一个弱点。

参考文献

[1] Princeton University. WordNet: a lexical database for English [EB/OL]. [2013-06-10]. http://wordnet.princeton.edu/wordnet/.

[2] AGIRRE E, EDMONDS P. Word sense disambiguation: algorithms and applications[M]. New Your: Springer, 2007: 12-15.

[3] ZOUAGHI A, MERHBENE L, ZRIGUI M. Combination of information retrieval methods with LESK algorithm for Arabic word sense disambiguation[J]. Artificial Intelligence Review, 2012, 38(4): 257-269.

[4] Lu Wenpeng, Huang Heyan, Zhu Chaoyong. Feature words selection for knowledge-based word sense disambiguation with syntactic parsing[J]. Przeglad Elektrotechniczny, 2012, 88(1b): 82-78.

[5] TEJADA-CÁLRCAMO J, CALVO H, GELBUKH A, et al. Unsupervised WSD by finding the predominant sense using

图1 算法示意图

context as a dynamic thesaurus[J].Journal of Computer Science and Technology, 2010, 25(5): 1030-1039.

[6] RAFA M, ADAM P.The WSD development environment[C]. Lecture Notes in Computer Science, 6562, 2011: 224-233.

[7] RIAHI N, SEDGHI F.A semi-supervised method for persian homograph disambiguation[C].2012 20th Iranian Conference on Electrical Engineering(ICEE 2012), Tehran, Iran: 2012: 748-751.

[8] HARRIS Z.Distributional structure[M].New York: Oxford University Press, 1985: 26-47.

[9] PANTEL P, Lin Dekang.Discovering word senses from text[C]. In Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Edmonton, Canada: 2002: 613-619.

[10] Han Lushan, FININ T, MCNAMEE P, et al.Improving word similarity by augmenting PMI with estimates of word polysemy[J].IEEE Transactions on Knowledge and Data Engineering, 2013, 25(6): 1307-1322.

[11] KAUR I, HORNOF A J.A comparison of LSA wordnet and PMI-IR for predicting user click behavior[C].In Proc. Human Factors in Computing Systems Conf, ACM Press, 2005: 51-60.

(收稿日期: 2013-08-27)

作者简介:

王永生, 男, 1972年生, 博士, 高级工程师, 主要研究方向: 计算语言学。