基于语义密度的名词消歧算法

何文垒 刘功申
(上海交通大学信息安全工程学院 上海 200240)

摘 要 提出了一种以概念相关性为主要依据的名词消歧算法。与现有算法不同的是,该算法在 WordNet 上对两个语义之间的语义距离进行了拓展,定义了一组语义之间的语义密度,从而量化了一组语义之间的相关性。将相关性转化为语义密度后,再进行消歧。还提出了一种在 WordNet 上的类似 LSH 的语义哈希,从而大大降低了语义密度的计算复杂度以及整个消歧算法的计算复杂度。在 SemCor 上对该算法进行了测试和评估。

关键词 消歧,名词消歧,语义密度,语义哈希
中图法分类号 TP18 **文献标识码** A

Noun Sense Disambiguation Based on Semantic Density

HE Wen-lei LIU Gong-shen
(School of Information Security Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract Proposed a novel approach for noun sense disambiguation based on concept correlation. Different from existing algorithms, we extended the notion of semantic distance on WordNet by defining a semantic density for a group of word senses, thus quantizing the correlation among a group of word senses. We disambiguated noun sense after converting the correlation into semantic density. Besides, we also proposed an LSH like semantic hashing on WordNet. With semantic hashing, we greatly reduced the time complexity of calculating semantic density and that of the whole disambiguation algorithm. Experiments and evaluation of this novel approach on SemCor were made.

Keywords Disambiguation, Noun sense disambiguation, Semantic density, Semantic hashing

1 引言

语义消歧一直以来都是自然语言处理中的一个难点。各种自然语言中普遍存在一词多义的现象,语义消歧就是要让计算机根据词所在的上下文来推断词在上下文中的语义(义项)。自然语言处理领域中,语义消歧是一个十分重要的环节,机器翻译、信息检索等都是以它为基础的。

目前的语义消歧方法主要分为以下几类[1]:

- 基于词典的方法
- 基于机器学习的方法
  - 监督学习
  - 无监督学习
- 混合方法

基于词典的消歧方法也称为规则消歧。Lesk[2]是基于词典算法的代表,它使用电子词典进行消歧。其他一些基于词典的算法则使用 WordNet[3,4]、平行语料库等不同的资源。监督学习算法[5]通过语义标注语料库将语义消歧问题转化为一个分类问题:词的义项代表不同的类,分类器将特定上下文中的词归为某一类,因此也称为统计消歧。相应地,各种分类算法都能应用到这个消歧算法中去,如朴素贝叶斯、神经网络、支持向量机等。无监督学习消歧算法不需要使用任何资源就能进行消歧,然而它不能对词语的语义进行严格的标注,只能通过词义将单词进行归类,将意义相近的词归为一组。因此,这种消歧算法和聚类算法非常相似。有些聚类算法[6]被应用到无监督消歧中,但是无监督消歧算法在自然语言处理中的应用却比其他算法少得多。混合算法则试图通过同时使用词汇库和语料库资源来提升消歧效果。

本文将介绍一种新的利用上下文语义关联的特点结合 WordNet 对名词进行语义消歧的方法。基于 WordNet 定义了一种语义密度。WordNet 是一种概念层次树状结构,其中的每一个节点又分别是一棵子树。本文所定义的一棵子树(一组同义词集)的语义密度,数值上与子树的大小负相关,与子树中所包含的上下文的语义正相关。这样定义的语义密度实际上就是语义关联性的一种反映,量化了一组同义词集之间的相关性。消歧时,选取语义密度最大的子树所包含的语义。为快速计算语义密度,本文还提出一种 WordNet 上的语义哈希,根据语义为不同的节点进行编码。通过语义哈希,同义词集之间的语义距离被直接量化为它们语义哈希的差。最后,在 SemCor[7]语义标注语料库上测试了算法的消歧效果。

本文第 2 节简要介绍 WordNet,并阐述了基于 WordNet 的语义哈希;第 3 节详细描述了我们定义在 WordNet 上的语义密度以及基于语义密度的消歧算法;第 4 节在 SemCor 语

到稿日期:2011-07-11 返修日期:2011-10-01 本文受 863 计划项目(2010AA012505),教育部科技发展中心项目(2010121)资助。
何文垒(1987—),男,硕士生,主要研究方向为自然语言处理,E-mail:aktoon@sjtu.edu.cn;刘功申(1974—),男,博士,副教授,主要研究方向为内容安全、自然语言处理。

料库上对本文所述的消歧算法进行了测试和评估。

2 WordNet 和语义哈希

WordNet 是由 Princeton 大学的认知科学实验室开发的一种机器可读词典(Machine-Readable Dictionary,MRD)。它是一个树状结构的词典,同义词集就是树中的节点。WordNet 中,每个同义词集(Synonym set,synset)代表一个语义或者概念。例如,WordNet 中,同义词集{car,auto,automobile,machine,motorcar}代表"由引擎驱动的四轮车辆(4-wheeled motor,usually propelled by an internal combustion engine)"的概念。WordNet 中父子节点的关系就是上位词与下位词的关系。节点的父节点就是节点的上位词,即比当前节点所代表的概念更一般、更宽泛、更抽象的同义词集;而节点的子节点就是节点的下位词,即比当前节点更具体的同义词集。例如,{educator,pedagogue}(someone who educates young people)是{teacher,instructor}(a person whose occupation is teaching)的上位词,即教育工作者是教师的上位词。Princeton 大学开发的 WordNet 是针对英语的,目前世界上许多国家都已经或正在开发和 Princeton WordNet 对齐的其他语言的 WordNet,如 ItalWordNet[8],EuroWordNet[9] 等。中文的 WordNet 有北大的中文概念词典[10](Chinese Concept Dictionary,针对简体中文)以及台湾的中文词汇网络[11](Chinese WordNet,针对繁体中文)。本文基于 Princeton 的英文 WordNet 来研究名词语义消歧的方法,但所述的消歧方法实际上与语言无关,即以同样的方法,选用中文 WordNet 便可对中文名词进行消歧。

本文的名词消歧算法是基于定义在 WordNet 上的语义密度来实现的。为了能快速地计算语义密度,我们希望将 WordNet 中的所有同义词集(synset)映射到一个数值空间,即对同义词集 s,有 $\varphi(s)=h$,h 为一整数。且 $\varphi(s)$满足:

1)对任意 $s_1\neq s_2$,有 $\varphi(s_1)\neq\varphi(s_2)$。

2)对任意 s_1,s_2,s_3,若 $dist(s_1,s_2)<dist(s_1,s_3)$,则 $\varphi(s_1)$,$\varphi(s_2)$,$\varphi(s_3)$满足 $\Pr(|\varphi(s_1)-\varphi(s_2)|<|\varphi(s_1)-\varphi(s_3)|)>1-\varepsilon,\varepsilon\to 0$。

其中 $dist(s_i,s_j)$为 s_i,s_j 所在节点间的最短路径的带权长度,即 $dist(s_i,s_j)=\sum_k weight(e_k)$,而对于边 e,其权重为 $weight(e)=2^l$,l 为由下至上数边 e 所在的层数。

显然 s_i,s_j 在 WordNet 中的意义差别越大,那么 $dist(s_i,s_j)$就越大。实际上 $dist(s_i,s_j)$就是 s_i,s_j 定义在 WordNet 上的一种语义距离。通过 $\varphi(s)$,就能很直接地将语义距离 $dist(s_i,s_j)$转化为 $\varphi(s)$的差。因为 $\varphi(s)$的作用与哈希函数类似,并且 $\varphi(s)$的值实际上体现了同义词集的语义信息,所以 $\varphi(s)$越接近则语义越相关。这一特点与 LSH 函数[12]类似,所以称 $\varphi(s)$为语义哈希。通过对 WordNet 的研究,我们发现直接将 WordNet 的树状结构编码到 128 位整数上,就能获得近似满足上述要求的语义哈希 $\varphi(s)$。WordNet 中的名词绝大部分都在以{entity}为根的树下(一些专有门词、地名、国家名等不在{entity}树下,这些名词通常也不是消歧的重点)。图 1 显示了{entity}树中每层的最大子节点数以及编码 1 到 K 层所需的二进制位数。我们用图 2 所示的方法,用 128 位整数来对 WordNet 中的名词同义词集逐层编码,从而得到 $\varphi(s)$,这样得到的语义哈希 $\varphi(s)$便能满足上述 2 个要求。我们将编码后每个名词同义词集与其语义哈希的对应关系记录下来,任意给出一个名词同义词集,查表即可快速得到其语义哈希值 $\varphi(s)$。语义距离是语义密度的基础,然而语义距离的计算并不简单。有了语义哈希之后,我们就能通过同义词集语义哈希之间的差来量化同义词集的语义距离。下面提到语义哈希或者 $\varphi(s)$时,均指本节所述的编码和 $\varphi(s)$函数。

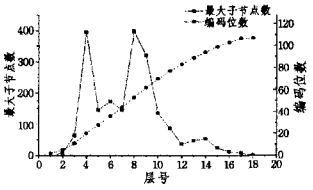

图 1 WordNet 语义哈希位数与层数关系

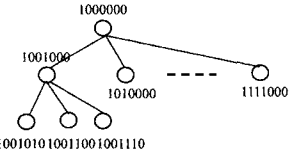

图 2 WordNet 语义哈希编码方法示意图

3 语义密度和名词语义消歧

概念相关性是名词消歧的一个重要依据。例如,一个词的某个义项与上下文中其他词的义项更相关,则该义项更有可能是当前上下文中的正确义项。这种相关性可以由 WordNet 上的语义距离来描述,如前面所述的 $dist(s_i,s_j)$。语义距离 $dist(s_i,s_j)$根据 WordNet 的层次结构为词与词之间的相关性提供了一个基准,量化了词与词之间的相关性。然而在消歧过程中,需要考虑上下文窗口中所有词的相关性。为此在语义距离的基础上,又定义了语义密度。语义密度量化了一组词之间的概念相关性。对于一组同义词集 $s_1,s_2,\cdots,s_n$,我们的语义密度 $density(s_1,s_2,\cdots,s_n)$定义为 n^3 与包含所有 s_1,$s_2,\cdots,s_n$ 的最小子树的"体积"$V_{\min}$的商,由下式给出:

$$density(s_1,s_2,\cdots,s_n)=\frac{n^3}{V_{\min}}=\frac{n^3}{1.2^{\max(b-72,0)}} \quad (1)$$

式中,b 是 $\varphi(s_1),\varphi(s_2),\cdots,\varphi(s_n)$的最低相同位的位置,例如 0010101b,0011101b,0010001b 的最低相同位 b 为 5。式(1)是经过多次实验和调整后得到的,能使语义消歧有较好的效果。观察式(1)不难发现,当 n 相同时,语义密度与最低相同位 b,即包含所有 $s_1,s_2,\cdots,s_n$ 的最小子树的"体积"负相关,"体积"越小则 $s_1,s_2,\cdots,s_n$ 相互的语义距离越小,相关性越强。而语义密度又与子树中包含的同义词集数目正相关。可见,语义密度很好地体现了一组同义词集的相关性,语义密度越大,则这组同义词集更为相关。

有了语义密度,就能方便地利用概念相关性来进行名词语义消歧了。从消歧上下文窗口中所有词的可能的义项中选取任意个,计算它们的语义密度,语义密度最大的子树所包含的义项即是消歧的结果。首先设定消歧上下文窗口的大小,

窗口中仅包含名词，所有词都是消歧算法的输入，最中间的词就是被消歧的词。例如，取窗口大小为 21，窗口为 $\{w_1, w_2, \cdots, w_{10}, w_{11}, w_{12}, \cdots, w_{20}, w_{21}\}$，那么 w_{11} 即为被消歧的词。每次完成一个词的消歧后，消歧窗口向右移动一个名词。上述窗口右移一个名词后为 $\{w_2, w_3, \cdots, w_{11}, w_{12}, w_{13}, \cdots, w_{21}, w_{22}\}$，此时 w_{12} 为被消歧的词。消歧窗口依次右移，直到所有的名词都完成消歧。

对于一个长度为 $2l+1$ 的消歧窗口 W：$\{w_k, w_{k+1}, \cdots, w_{k+l-1}, w_{k+l}, w_{k+l+1}, \cdots, w_{k+2l-1}, w_{k+2l}\}$，被消歧词为 w_{k+l}，具体消歧算法如下：首先将 W 中每个词的候选同义词集合并成一个大的候选集 C，即假如包含 w_i 的同义词集有 $s_{i_1}, s_{i_2}, \cdots$，记为 $synset(w_i)=\{s_{i_1}, s_{i_2}, \cdots\}$，则 $C=\bigcup_{i=k}^{k+2l} synset(w_i)$。得到候选集 C 后，将 C 中的所有同义词集按语义哈希值 $\varphi(s)$ 排序，得到 $C=\{s_1, s_2, \cdots, s_p\}$，其中 $\varphi(s_1)\leqslant\varphi(s_2)\leqslant\cdots\leqslant\varphi(s_p)$。接下来，需要计算 C 中任意几个同义词集的语义密度，并选取其中语义密度最大的子树下的同义词集作为消歧结果。根据语义哈希的特性，显然在排好序的 $\{s_1, s_2, \cdots, s_p\}$ 中选择连续的同义词集能获得更大的语义哈希，因此为获得语义密度的最大值，我们只考虑连续的同义词集。然而，若选取的连续同义词集 C'：$\{s_i, s_{i+1}, \cdots, s_{i+m}\}$ 与被消歧词 w_{k+l} 的候选同义词集 $synset(w_{k+l})$ 没有交集，那么这样的连续同义词集显然对消歧 w 是没有意义的。因此，在计算候选集 C 中任意几个同义词集的语义密度时，我们只考虑和被消歧词 w_{k+l} 的候选同义词集 $synset(w_{k+l})$ 有交集的且哈希值连续的同义词集。计算完所有的语义密度后，选取语义密度最大的连续的同义词集，认为其中与 $synset(w_{k+l})$ 共有的同义词集便是被消歧词 w_{k+l} 在当前上下文中的语义。在一些情况下，这样得到的同义词集并不唯一，此时取语义哈希值接近 C' 中语义哈希中位数的同义词集作为消歧结果。至此，便完成了对 w_{k+l} 的消歧。然后将消歧窗口右移一个名词，重复上述过程，直至所有名词都完成消歧。上述消歧算法可由下面的伪代码来表示，其中，$sense(w)$ 代表消歧结果。

```
while(has noun to disambiguate)
maxDensity←0
R←Ø
W←{w_k, w_{k+1}, ..., w_{k+l-1}, w_{k+l}, w_{k+l+1}, ..., w_{k+2l-1}, w_{k+2l}}
C←⋃_{i=k}^{k+2l} synset(w_i)
C={s_1, s_2, ..., s_p}←SortBySemanticHash(C)
for (i=1→p-1)
  for(j=i+1→p)
    C'←{s_i, s_{i+1}, ..., s_j}
    if (1≤|C'∩synset(w_{k+l})|≤2)
      Density←density(C')
      if (Density>MaxDensity)
        MaxDensity←Density
        R←C'∩synset(w_{k+l})
sense(w_{k+l})←R
k←k+1
```

上述消歧算法中，充分利用语义哈希函数 $\varphi(s)$ 来提高消歧的效率。通过语义哈希，可以在 $O(n)$ 时间内估算出 n 个同义词集的最低公共父节点的位置，这降低了计算语义密度的时间复杂度。按语义哈希排序候选同义词集 C，我们又将计算语义密度的次数从 $O(2^p)$ 降低到 $O(p^2)$。

4 实验结果

我们在 SemCor2.1 上测试了本文所述名词消歧算法的效果。SemCor 是一个公开的语义标注语料库，其中的文档来自 Brown 语料库。SemCor 中的单词语义是基于 WordNet 标注的。我们选择了 SemCor 中的一些文档，用 TreeTagger[13] 对它们进行 POS 标注，然后将 TreeTagger 标注的结果作为消歧算法的输入进行消歧。最后将算法的消歧结果与 SemCor 中人工标注的结果进行对比，得到消歧算法的准确率。表 1 列出了窗口大小为 9 时 SemCor 中前 12 篇文档的消歧准确率。作为对照，表中基准列为随机猜测时消歧的准确率。例如，一个词有 4 个同义词集时，随机猜测的准确率为 25%。

表 1 消歧准确率

文档	准确率	基准
br-a01	61.71%	28.64%
br-a02	57.32%	25.99%
br-a11	44.09%	24.46%
br-a12	49.15%	24.61%
br-a13	37.50%	23.67%
br-a14	57.35%	26.78%
br-a15	55.65%	25.84%
br-b13	49.72%	24.62%
br-b20	50.71%	26.09%
br-c01	47.77%	24.92%
br-c02	45.19%	22.07%
br-c04	44.87%	22.51%

消歧算法在 SemCor 上的平均准确率是 49%，好于同属于规则消歧的 Lesk[1]（36%）和文献[14]中的算法（44%）。文献[14]中的算法与本文的算法都将概念相关性作为消歧的依据，然而由于本文算法引入了语义哈希，因此计算复杂度要比文献[14]中的算法低很多。

图 3 显示了窗口大小对算法消歧准确率的影响。可见，在一定范围内，窗口大小对准确率影响不是很大；而当窗口太大时，准确率反而会下降，因为此时窗口中包含了与被消歧词不太相关的内容，干扰了消歧。图 4 显示了消歧平均准确率与被消歧词多义度的关系。

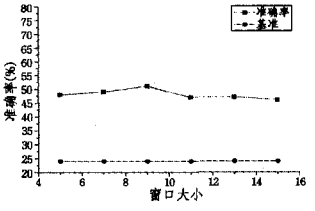

图 3 窗口大小对消歧准确率的影响

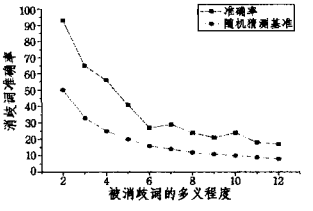

图 4 被消歧词多义程度与消歧准确率的关系

结束语 本文提出了一种新的利用概念相关性来进行名词消歧的算法。提出了 WordNet 上的语义哈希，并在 WordNet 基础上定义了一种语义密度，量化了一组同义词集之间的相关性。算法把语义密度作为消歧的依据，并通过语义哈

希大大减少了消歧时语义密度的计算次数，从而降低了消歧的计算复杂度。最后在 SemCor 语料库上对本文提出的算法进行了测试。结果显示，本文提出的算法与同类算法相比有较好的性能。然而本文的算法仅针对名词消歧，如何处理名词以外的词以及 WordNet 中没有登记的词，则是本文所述算法没有解决的问题，有待进一步研究和改进。

参 考 文 献

[1] Turdakov D Y. Word Sense Disambiguation Methods [J]. Programming and Computer Software, 2010, 36(6): 309-326

[2] Lesk M. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone [C]// Proceedings of the 5th Annual International Conference on Systems Documentation (SIGDOC '86). New York, NY, USA: ACM, 1986: 24-26

[3] Miller G A. Wordnet: A Lexical Database for English[J]. Communications of the ACM, 1995, 38(11): 39-41

[4] Cao D D, Basili R, Luciani M, et al. Robust and Efficient Page Rank for Word Sense Disambiguation[C]// Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing. Uppsala, Sweden: ACL, 2010: 24-32

[5] Liu Hong-fang, Teller V, Friedman C. A Multi-aspect Comparison Study of Supervised Word Sense Disambiguation[J]. Journal of the American Medical Informatics Association, 2004, 11(4): 320-331

[6] Berkhin P. Survey of Clustering Data Mining Techniques[R]. CA, USA: Accrue Software, Inc., 2002

[7] Miller G A, Leacock C, Tengi R, et al. A Semantic Concordance [C]// Proceedings of the Workshop on Human Language Technology (HLT'93). Morristown, NJ, USA: Association for Computational Linguistics, 1993: 303-308

[8] Roventini, Alonge, Bertagna, et al. ItalWordNet: Building a Large Semantic Database for the Automatic Treatment of Italian[J]. Linguistica Computazionale, 2003, 18: 745-791

[9] Vossen P. EuroWordNet: building a multilingual database with wordnets for European languages[J]. The ELRA Newsletter, 1998, 3(1): 7-10

[10] 于江生，俞士汶. 中文概念词典的结构[J]. 中文信息学报，2002，16(4)：12-20

[11] 黄居仁，谢舒凯，洪嘉馡，等. 中文词汇网络：跨语言知识处理基础架构的设计理念与实践[J]. 中文信息学报，2010，24(2)：14-23

[12] Ravichandran D, Pantel P, Hovy E. Randomized algorithms and NLP: using locality sensitive hash function for high speed noun clustering[C]// Proceedings of the 43rd Annual Meeting of the ACL. Ann Arbor, USA: ACL, 2005: 622-629

[13] Schmid H. Probabilistic Part-of-Speech Tagging Using Decision Trees[C]// Proceedings of the International Conference on New Methods in Language Processing. Manchester, UK, 1994

[14] Agirre E, Rigau G. Word sense disambiguation using conceptual density[C]// Proceedings of the 16th Conference on Computational linguistics. Morristown, NJ, USA, 1996: 16-22

(上接第 193 页)

集变小。当删除单个对象时，如果删除的对象不属于集合 X，则集合 X 的上近似集变小，下近似集变大；如果删除的对象属于集合 X，则集合 X 的上、下近似集都变小。最后提出了动态增量更新算法，并给予了实验验证。对于对象集和属性集同时变化的研究将另文介绍。

参 考 文 献

[1] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Sciences, 1982, 11: 341-356

[2] Ziarko W. Variable precision rough set model [J]. Journal of Computer and System Sciences, 1993, 46: 39-59

[3] Wang J, Xu L. Probabilistic rough set models[J]. Computer Science, 2002, 29(8): 76-78

[4] Nanda S, Majumdar S. Fuzzy rough sets[J]. Fuzzy Sets and Systems, 1992, 45(2): 157-160

[5] Stefanowski J, Tsoukias A. Incomplete information tables and rough classification[J]. Computational Intelligence, 2001, 17: 54-566

[6] Kryszkiewicz M. Rough set approach to incomplete information systems[J]. Information Sciences, 1998, 112: 39-49

[7] Grzymala-Busse J W. Data with missing attribute values: generalization of indiscernibility relation and rule induction[J]. Transactions on rough sets I, Lecture Notes in Computer Science, Springer-Verlag, 2004, 3100: 78-95

[8] Grzymala-Busse J W. Characteristic relations for incomplete data: a generalization of the indiscernibility relation[C]// Proceedins of the Third International Conference on Rough Sets and Current Trends in Computing (RSCTC 2004). Lecture Notes in Artificial Intelligence, Springer-Verlag, 2004, 3066: 244-253

[9] 刘伟斌，李天瑞，邹维丽，等. 特性关系粗糙集下属性值粗化细化时近似集增量更新方法研究[J]. 计算机科学，2010，37(6)：248-251

[10] Grzymala-Busse J W. A new version of the rule induction system LERS[J]. Fundamenta Informaticae, 1997, 31: 27-39

[11] Slowinski R, Vanderpooten D. A generalized definition of rough approximations based on similarity[J]. IEEE Transactions on Knowledge and Data Engineering, 2000, 12: 331-336

[12] Li T, Ruan D, Geert W, et al. A rough sets based characteristic relation approach for dynamic attribute generalization in data mining[J]. Knowledge-based Systems, 2007, 20(5): 485-494

[13] Song J, Li T, Ruan D. An integration of cloud transform and rough set theory to induction of decision trees[J]. Fundamenta Informaticae, 2009, 94(2): 261-273