18. 单词意义和 WordNet

> **Lady Bracknell**. Are your parents living?
> **Jack**. I have lost both my parents.
> **Lady Bracknell.** To lose one parent, Mr. Worthing, may be regarded as a misfortune; to lose both looks like carelessness.
>
> Oscar Wilde, *The Importance of Being Earnest*

单词是有**歧义(ambiguous)**的:同一个词可以用来表示不同的东西。在第六章中，我们看到“mouse”这个词(至少)有两种意义:(1)一种小型啮齿动物，或者(2)一种手动控制光标的装置。“bank”这个词的意思是:(1)一个金融机构;(2)一个倾斜的堤岸。在上面的戏剧《认真的重要性》中，奥斯卡·王尔德扮演了“失去”的两种意义(放错地方的东西和遭受失去一个人的痛苦)。

我们说，mouse 或 bank 是**一词多义的(polysemous)**(源自希腊语 have many senses，poly 表示 many, sema 表示 sign 或 mark)[1]。**意义(sense)**(或词义)是一个单词的含义的一个方面(aspect)的离散表示。在本章中，我们将更详细地讨论词义，并介绍 WordNet，一个大型的在线**词典**(thesauruses)--表示词义的数据库--具有多种语言的版本。WordNet 也代表了意义之间的关系。例如，在狗和哺乳动物之间存在一个 IS-A 关系(狗是一种哺乳动物)，在引擎和汽车之间存在一个部分-整体关系(引擎是汽车的一部分)。

了解两个意义之间的关系在语言理解中起着重要的作用。考虑一下反义关系(antonymy)。如果两个词有相反的含义，那它们就是反义词，比如 long 和 short, up 和 down。区分这些元素对于语言理解非常重要(如果用户要求对话施事者将音乐调大，则不幸的是调低音乐)。但事实上，在 word2vec 这样的嵌入模型中，反义词很容易相互混淆，因为在嵌入空间中与单词最接近的单词(如 up)往往是它的反义词(如 down)。代表此关系的词典会有所帮助!

我们还引入了词义消歧(word sense disambiguation，WSD)，即确定在特定上下文中使用的单词的意义。我们将给出有监督的和无监督的算法来决定在特定的环境下意图使用哪种意义。这项任务在计算语言学和许多应用中有很长的历史。在回答问题时，如果我们知道哪一种 bat 意义是相关的，则我们可以对询问“bat care”的用户更有帮助。(用户是吸血鬼(vampire)吗?或者只是想打棒球(baseball)。)一个词的不同意义通常会有不同的翻译;在西班牙语中，动物 bat(蝙蝠)是一种 murcielago，而 baseball bat(棒球棒)是一种 bate，确实，词义算法可以帮助改进 MT (Pu 等人，2018)。最后，WSD 长期以来一直被用作评估自然语言理解模型的工具，而理解模型如何表示不同的词义是一个重要的分析方向。

18.1. 单词意义

意义是一个单词的含义的一个方面的离散表示。宽松地遵循词典编纂的传统，我们用上标来表示每种意义：bank[1] 和 bank[2]，mouse[1] 和 mouse[2]。在上下文中，很容易看到不同的含义：

mouse[1]:…1968 年，**鼠标**控制电脑系统。
mouse[2]:…像**老鼠**一样安静的动物
bank[1]:…**银行**可以将投资存放在托管账户中。
bank[2]:…随着东**岸**农业的迅速发展，这条河…

18.1.1. 定义单词意义

我们如何定义单词意义的含义(the meaning of a word sense)?我们在第六章介绍了将一个单词表示为语义空间中的一个嵌入点的标准计算方法。像 word2vec 或 GloVe 这样的嵌入模型的直觉是，一个单词

[1] 你可能还会看到**多义关系（polysemy)**的不同用法，指的是一个词的意思有某种语义关系的情况，而**同形关系（homonymy)**则用于意思之间没有关系的情况。

的意义可以通过它的共现来定义，即经常在附近出现的单词的数量。但这并没有告诉我们如何定义单词意义的含义。正如我们在第 10 章中看到的，像 BERT 这样的上下文嵌入更进一步，它提供了一个在文本上下文中表示单词含义的嵌入，我们将看到上下文嵌入是现代词义消歧算法的核心。

但首先，我们需要考虑**字典(dictionary)**和**词典(thesaurus)**提供的另一种定义意义的方式。一个是基于这样的事实，字典或词典给每一种意义的文本定义称为**注释(gloss)**。以下是 bank 的两种意义的注释:

1. financial institution that accepts deposits and channels the money into lending activities
2. sloping land (especially the slope beside a body of water)

注释并不是一种正式的含义表示;它们只是为人类而写的。请参考 American Heritage Dictionary (Morris, 1985)中的关于 right, left, red, 和 blood 的定义。

right	adj.	located nearer the right hand esp. being on the right when facing the same direction as the observer.
left	adj.	located nearer to this side of the body than the right.
red	n.	the color of blood or a ruby.
blood	n.	the red liquid that circulates in the heart, arteries and veins of animals.

注意这些定义中的**回环(circularity)**。right 的定义直接指向了它自己，而 left 的词条在短语 this side of the body 中包含了一个隐含的自我引用，这大概意味着 left。red 和 blood 的词条在定义上相互参照。对于人类来说，这样的词条是有用的，因为字典的用户对这些其他术语有充分的了解。

然而，尽管它们具有回环并且缺乏正式表示，但是注释仍然可以用于意义的计算建模。这是因为注释只是一个句子，我们可以从句子中计算出句子的嵌入来告诉我们一些关于意义的东西。字典经常会提供例句和注释，这些可以再次用来帮助建立一种意义的表示。

词典为定义一种意义提供的第二种方式是像字典定义一样，通过其与其他意义的关系来定义一种意义。例如，以上定义清楚地表明，right 和 left 是相似的词元，彼此之间存在某种交替或对立。同样，我们可以收集到 red 是一种颜色，而 blood 是一种液体。这种意义上的关系(IS-A 或反义词)在 WordNet 等在线数据库中明确列出。给定足够大的此类关系数据库，许多应用程序都具有执行关于词义的复杂语义任务的能力(即使它们并不真正从 left 知道自己的 right)。

18.1.2. 单词有多少种意义?

字典和词典给出离散的意义列表。相比之下，嵌入(无论是静态的还是上下文的)提供了一个连续的高维意义模型，它不会分解为离散的意义。

因此，创建词典取决于用于确定何时应以离散的意义来表示单词的不同用法的标准。如果两个意义具有独立的真实条件，不同的句法行为和独立的意义关系，或者它们表现出对立的意义，我们可能会认为它们是离散的。

考虑一下 WSJ 语料库中的动词 serve 的以下用法：

(18.1) They rarely serve red meat, preferring to prepare seafood.

(18.2) He served as U.S. ambassador to Norway in 1976 and 1977.

(18.3) He might have served his time, come out and led an upstanding life.

serving red meat 中的 serve，以及 serving time 中的 serve，显然具有不同的真值条件和预先设定。serve as ambassador 中的 serve 具有作为 NP 的独特子类结构。这些启发意味着可能有三种截然不同的 serve 意义。判定两种意义是否不同是一种实用的技术，该技术是将一个单词的两种用法合成到一个句子中。这种将相反意义的阅读知识(readings)结合起来的方法称为**轭合(zeugma)**。考虑以下例子:

(18.4) Which of those flights serve breakfast?

(18.5) Does Air France serve Philadelphia?

(18.6) ?Does Air France serve breakfast and Philadelphia?

我们用(?)来标记那些语义上有错误格式的例子。虚构的第三个例子(一个轭合案例)的奇怪之处表明，没有一种明智的方法可以使得 serve 的单独意义同时适用于 breakfast 和 Philadelphia。我们可以以此为

依据来证明在这种情况下 serve 有两种不同的意义。

字典倾向于使用许多细粒度的意义来捕捉细微的含义差异，考虑到字典的传统作用是帮助单词学习者，这是一种合理的方法。为了计算的目的，我们通常不需要这些细微的区别，所以我们经常将意义进行分组或聚类;我们已经在本章的一些例子中这样做了。实际上，将示例聚类为意义，或将意义聚类为更粗粒度的类别，是一项重要的计算任务，我们将在第 18.7 节中讨论。

18.2. 意义之间的关系

本节探讨意义之间的关系，特别是那些已经接受了大量计算研究的单词，如**同义关系(synonymy)**、**反义关系(antonymy)**和**上位关系(hypernymy)**。

同义关系

我们在第六章介绍过，当两个不同词(词元的两个意义相同或几乎相同时，我们说这两个意义是**同义词(synonym)**。同义词包括这样的对：

couch/sofa vomit/throw up filbert/hazelnut car/automobile

我们也提到过，在实践中，同义词这个词通常用来描述一种近似或粗略的同义词关系。此外，同义词实际上是一种意义之间的关系，而不是单词之间的关系。考虑单词 big 和 large。这两个词在下面的句子中看起来可能是同义词，因为我们可以在两个句子中互换 big 和 large，而且保留相同的含义:

(18.7) How big is that plane?

(18.8) Would I be flying on a large or small plane?

但是请注意下面的句子，我们不能用 large 来代替 big:

(18.9) Miss Nelson, for instance, became a kind of big sister to Benjamin.

(18.10) ?Miss Nelson, for instance, became a kind of large sister to Benjamin.

这是因为“big”这个词有“变老”或“长大”的意思，而“large”却没有这个意思。因此，我们说 big 和 large 的某些意义(几乎)是同义的，而其他的意义则不是。

反义关系

同义词是具有相同或相似意义的单词，而**反义词(antonym)**是具有相反意义的单词，例如：

long/short big/little fast/slow cold/hot dark/light rise/fall up/down in/out

如果两种意义定义了一种二元对立或在某种程度上处于对立的两端，那么它们可以是反义词。这就是长/短，快/慢，或大/小的情况，它们在长度或尺寸尺度的两端。另一组反义词，**可逆词(reversives)**，描述在相反方向上的变化或运动，如 rise/fall 或 up/down。

因此，反义词在一个方面的意义完全不同—它们在尺度上或方向上的位置—但在其他方面却非常相似，几乎所有其他方面的意义都相同。因此，自动区分同义词和反义词可能很困难。

分类关系

另一种联系词义的方法是**分类(taxonomically)**。一个词(或意义)是另一个词或意义的**下位词(hyponym)**，如果前者更具体，表示另一个词或意义的子类。例如 car 是 vehicle 的下位词，dog 是 animal 的下位词，mango 是 fruit 的下位词。反之，我们说 vehicle 是 car 的**上位词(hypernym)**，animal 是 dog 的上位词。不幸的是，这两个词(hypernym 和 hyponym)非常相似，因此很容易混淆;由于这个原因，**superordinate** 这个词经常被用来代替 hypernym。

Superordinate	vehicle	fruit	furniture	mammal
Subordinate	car	mango	chair	dog

我们可以更正式地定义上位关系，即由上位词表示的类外延地包含由下位词表示的类。因此，动物的类别包括所有的狗，而移动(moving)动作的类别包括所有的行走(walking)动作。上位关系也可以根据**蕴涵(entailment)**来定义。在此定义下，如果 A 的所有对象也都是 B，则意义 A 是意义 B 的下位词，因此，A 蕴涵 B 或$\forall x \quad A(x) \Rightarrow B(x)$。

下位关系/上位关系通常是一种传递关系。如果 A 是 B 的下位词，而 B 是 C 的下位词，则 A 是 C 的下

位词。

上位词/下位词结构的另一个名称是 IS-A 层次结构，其中我们说 A IS-A B 或 B 包含(subsumes)A。

上位关系在文本蕴涵或问题回答等任务中很有用；例如，知道白血病(leukemia)是一种癌症，在回答有关白血病的问题时肯定会很有用。

部份-整体关系

另一种常见关系是 **meronymy**，即**部分-整体(part-whole)**关系。腿是椅子的一部分;车轮是汽车的一部分。我们说 wheel 是 car 的**部分词(meronym)**, car 是 wheel 的**整体词(holonym)**。

结构化的一词多义

一个单词的意义也可以在语义上有联系，在这种情况下，我们称之为结构化一词多义(structured polysemy)。想想这个 bank 的意义:

(18.11) The bank is on the corner of Nassau and Witherspoon.

这个意义，可能是 bank[4]（银行），含义类似于“属于金融机构的建筑物”。这两种意义(组织和与组织相关的建筑)同时出现在许多其他词中(school, university, hospital 等)。因此，我们所代表的意义之间存在着一种系统的关系：

BUILDING ↔ ORGANIZATION

这种特殊的一词多义关系被称为隐喻(metonymy)。隐喻是使用一个概念或实体的一个方面来指实体的其他方面或实体本身。当我们用“白宫”这个短语来指办公室在白宫的政府时，我们就是在进行隐喻。其他常见的隐喻例子包括以下意义的成对关系:

AUTHOR *(Jane Austen wrote Emma)*	↔	**WORKS OF AUTHOR** *(I really love Jane Austen)*
FRUITTREE *(Plums have beautiful blossoms)*	↔	**FRUIT** *(I ate a preserved plum yesterday)*

18.3. WordNet：词汇关系数据库

在英语和许多其他语言中，最常用的语义关系资源是 **WordNet** 词汇数据库(Fellbaum, 1998)。英语 WordNet 由三个独立的数据库组成，分别是名词和动词，第三个是形容词和副词；不包括封闭类的单词。每个数据库都包含一组词元，每个词元都有一组语义注释。WordNet 3.0 版本有 117,798 个名词，11529 个动词，22,479 个形容词和 4,481 个副词。平均每个名词有 1.23 个意义，平均每个动词有 2.16 个意义。WordNet 可以从网上访问，也可以从本地下载。图 18.1 显示了名词和形容词 bass 的词元条目。

The noun “bass” has 8 senses in WordNet.
1. bass[1] - (the lowest part of the musical range)
2. bass[2], bass part[1] - (the lowest part in polyphonic music)
3. bass[3], basso[1] - (an adult male singer with the lowest voice)
4. sea bass[1], bass[4] - (the lean flesh of a saltwater fish of the family Serranidae)
5. freshwater bass[1], bass[5] - (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
6. bass[6], bass voice[1], basso[2] - (the lowest adult male singing voice)
7. bass[7] - (the member with the lowest range of a family of musical instruments)
8. bass[8] - (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

图 18-1：名词 bass 的 WordNet 3.0 词条的一部分

请注意，名词有八种意义，形容词有一种意义，每一种意义都有**注释(gloss)**(字典式的定义)，这个意义的同义词列表，有时也有用法示例(如形容词意义所示)。WordNet 没有表示发音的信息，因此不区分 bass[4]、bass[5] 和 bass[8] 中的[b ae s]发音与其他发音[b ey s]的意义。

WordNet 意义的**近义词(near-synonym)**集称为**同义词集(synset)**；在 WordNet 中，同义词集是一个重要的原语。bass 的词条包括{bass[1], deep[6]}或{bass[6], bass voice[1], basso[2]}这样的同义词集。我们可以把同义词集看作是我们在第 15 章中讨论过的类型的概念。因此，WordNet 不是用逻辑术语来表示概念，而是将它们表示为可以用来表达概念的词义列表。下面是另一个同义词集的示例:

{chump[1], fool[2], gull[1], mark[9], patsy[1], fall guy[1],sucker[1], soft touch[1], mug[2]}

这个同义词集的注释描述如下:

Gloss: a person who is gullible and easy to take advantage of.

(容易受骗和容易被利用的人。)

注释是一个同义词集的属性，因此包含在同义词集之中的每个意义都有相同的注释，可以表达这个概念。因为它们共享语义，像这样的同义词集是与 WordNet 词条相关联的基本单位，因此，在 WordNet 中参与大多数词汇语义关系的是同义词集，而不是词形、词元或个体意义。

WordNet 还用从语义字段中抽取的词汇分类来标注每个同义词集，例如，图 18.2 所示的名词有 26 个类别，动词有 15 个类别(加上形容词的 2 个类别和副词的 1 个类别)。这些类别通常被称为**超意义(supersense)**，因为它们充当粗糙的语义类别或意义分组，当词义过于细粒度时，这可能是有用的(Ciaramita 和 Johnson 2003, Ciaramita 和 Altun 2006)。超意义也被定义为形容词(Tsvetkov 等人，2014)和介词(Schneider 等人，2018)。

Category	Example	Category	Example	Category	Example
ACT	*service*	GROUP	*place*	PLANT	*tree*
ANIMAL	*dog*	LOCATION	*area*	POSSESSION	*price*
ARTIFACT	*car*	MOTIVE	*reason*	PROCESS	*process*
ATTRIBUTE	*quality*	NATURAL EVENT	*experience*	QUANTITY	*amount*
BODY	*hair*	NATURAL OBJECT	*flower*	RELATION	*portion*
COGNITION	*way*	OTHER	*stuff*	SHAPE	*square*
COMMUNICATION	*review*	PERSON	*people*	STATE	*pain*
FEELING	*discomfort*	PHENOMENON	*result*	SUBSTANCE	*oil*
FOOD	*food*			TIME	*day*

图 18-2 ：超意义:WordNet 中名词的 26 个词汇分类

18.3.1. WordNet 中的意义关系

WordNet 表示上一节讨论的所有类型的意义关系，如图 18.3 和图 18.4 所示。

Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	*breakfast*[1] → *meal*[1]
Hyponym	Subordinate	From concepts to subtypes	*meal*[1] → *lunch*[1]
Instance Hypernym	Instance	From instances to their concepts	*Austen*[1] → *author*[1]
Instance Hyponym	Has-Instance	From concepts to their instances	*composer*[1] → *Bach*[1]
Part Meronym	Has-Part	From wholes to parts	*table*[2] → *leg*[3]
Part Holonym	Part-Of	From parts to wholes	*course*[7] → *meal*[1]
Antonym		Semantic opposition between lemmas	*leader*[1] ⟺ *follower*[1]
Derivation		Lemmas w/same morphological root	*destruction*[1] ⟺ *destroy*[1]

图 18-3：WordNet 中的一些名词关系

Relation	Definition	Example
Hypernym	From events to superordinate events	*Fly*[9] → *travel*[5]
Troponym	From events to subordinate event	*walk*[1] → *stroll*[1]
Entails	From verbs (events) to the verbs (events) they entail	*snore*[1] → *sleep*[1]
Antonym	Semantic opposition between lemmas	*increase*[1] ⟺ *decrease*[1]

图 18-4：WordNet 中的一些动词关系

例如，WordNet 通过直接上位词和下位词的关系将每个同义词集与其直接更通用和更具体的同义词集相关联来表示下位关系(第 18.2 节)。可以遵循这些关系以产生具有更一般或更具体的同义词集的更长链。

图 18.5 显示了 bass[3] 和 bass[7] 的上位词链；更一般的同义词集在连续缩进的线上显示。

bass[3], basso (an adult male singer with the lowest voice)
=> singer, vocalist, vocalizer, vocaliser
=> musician, instrumentalist, player
=> performer, performing artist
=> entertainer
=> person, individual, someone...
=> organism, being
=> living thing, animate thing,
=> whole, unit
=> object, physical object
=> physical entity
=> entity

bass[7] (member with the lowest range of a family of instruments)
=> musical instrument, instrument
=> device
=> instrumentality, instrumentation
=> artifact, artefact
=> whole, unit
=> object, physical object
=> physical entity
=> entity

图 18-5：词元 bass 的两种不同意义的下位关系链

图注：请注意，这些链是完全不同的，只是会聚于非常抽象的层次 whole、unit。

WordNet 有两种分类实体:类和实例。实例是一个个体，一个专有名词，是一个独特的实体。例如，旧金山就是城市的一个例子。但城市是一个类，是自治市乃至地理位置的下位词。图 18.6 显示了 WordNet 的一个子图（(数据来自 Navigli 2016)），展示了许多关系。

18.4. 词义消歧

为一个词选择正确的意义的任务称为**词义消歧(word sense disambiguation)**，简称 WSD。WSD 算法以上下文中的一个词和一个固定的潜在词义清单作为输入，并输出上下文中正确的词义。

18.4.1. WSD:任务和数据集

在本节中，我们将介绍 WSD 的任务设置，然后转向算法。意义标记清单取决于任务。对于从英语翻译到西班牙语的上下文中的意义标记，英语单词的意义标记清单可能是不同西班牙语翻译的集合。对于医学文章的自动索引，意义标记清单可以是医学主题标题(Medical Subject Headings，MeSH)词典的词条集。或者，我们可以使用来自资源(如 WordNet)的意义集；或者，如果我们想要一个较粗的粒度集，则可以使用超意义(supersenses)。图 18.7 显示了单词 bass 的一些这样的示例。

在某些情况下，我们只需要消除一小部分单词的歧义。在这样的**词汇样本(lexical sample)**任务中，我们有一小部分预先选定的目标词和一些词汇表的意义清单。由于词汇集合和语义集合都很小，简单的监督分类方法工作得很好。

然而，更常见的是，我们有一个更难的问题，我们必须消除某些文本中所有单词的歧义。在这个**全词(all-words)**任务中，系统被给予一个完整文本和一个词汇表，其中每个词条都有一个意义清单，我们必须消除文本中的每个词的歧义(有时只是每个内容词)。全词任务类似于词类标注，不同之处在于它有一个更大的标记集，因为每个词元都有自己的标记集。这个更大的标记集的结果是数据稀疏性。

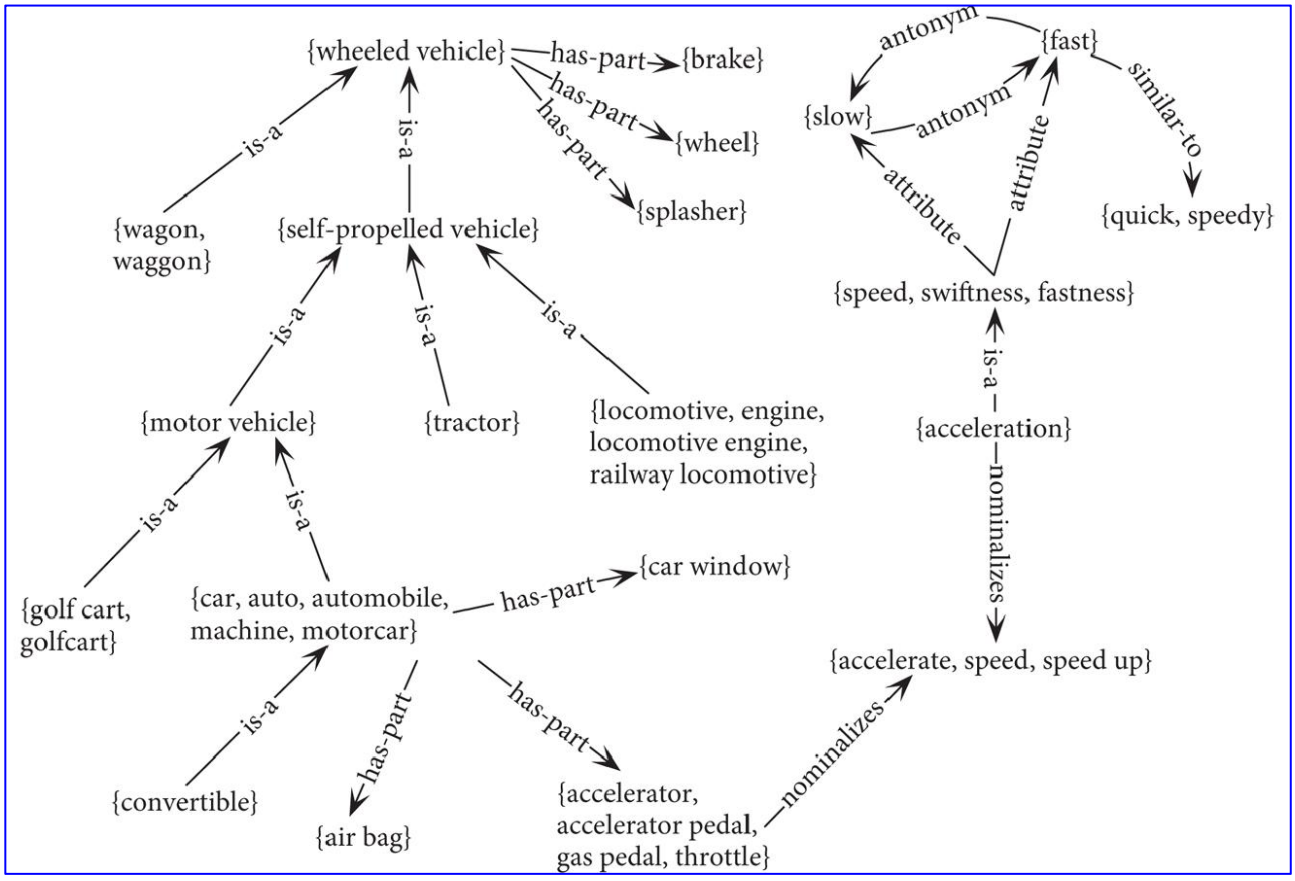

图 18-6：WordNet 作为一个图

WordNet Sense	Spanish Translation	WordNet Supersense	Target Word in Context
bass^4	lubina	FOOD	. . . fish as Pacific salmon and striped bass and. . .
bass^7	bajo	ARTIFACT	. . . play bass because he doesn't have to solo. . .

图 18-7：单词 bass 的一些可能的意义标记清单

受监督的全词消歧任务通常从**语义索引(semantic concordance)**(语料库)中进行训练，在这个语料库中，每个句子中的每个开放类词都用特定字典或词典(通常是 WordNet)中的词义进行标记。SemCor 语料库是 Brown 语料库的一个子集，由超过 226,036 个单词组成，这些单词是用 WordNet 意义手工标记的(Miller 等人 1993, Landes 等人 1998)。已经为 SENSEVAL 和 SemEval WSD 任务建立了其他意义标记的语料库，例如 SENSEVAL-3 Task 1 英语全词测试数据，包含 2282 个标注(Snyder 和 Palmer, 2004)或 SemEval-13 Task 12 数据集。大型语义索引在其他语言中也可用，包括荷兰语(Vossen 等人，2011)和德语(Henrich 等人，2012)。

这里有一个来自 SemCor 语料库的例子，显示了标记单词的 WordNet 意义数量;我们已经使用了标准的 WSD 表示法，其中一个下标标记了词类(Navigli, 2009):

(18.12) You will find$^{9}_{v}$ that avocado$^{1}_{n}$ is$^{1}_{v}$ unlike$^{1}_{j}$ other$^{j}_{j}$ fruit$^{1}_{n}$ you have ever$^{1}_{r}$ tasted$^{2}_{v}$

给定手工标记的测试集(如 fruit)中的每个名词、动词、形容词或副词，基于 semcord 的 WSD 任务是从 WordNet 中可能的意义中选择正确的意义。对于 fruit(水果)来说，这意味着要在正确答案“fruit^1_n”(种子植物成熟的生殖体)和其他两种意义的“fruit^2_n”(yield;产品的数量)和“fruit^3_n”(一些努力或行动的结果)之间作出选择。图 18.8 勾画了这个任务。

WSD 系统通常在本质上进行评估，通过计算 F1 来针对(against)在一个不提供的集合中手工标记的意义标记，例如上面讨论的 SemCor 语料库或 SemEval 语料库。

从标签语料库中为每个单词选择**最常见的意义(most frequent sense，MFS)**是一个令人惊讶的强大基线(Gale 等人，1992a)。对于 WordNet，这与第一个意义相对应，因为 WordNet 中的意义通常根据其

在 SemCor 意义标记语料库中的计数从最常见到最不常见排序。最常见的意义基线可以非常准确，因此经常用作默认值，在监督算法没有足够的训练数据时提供一个单词的意义。

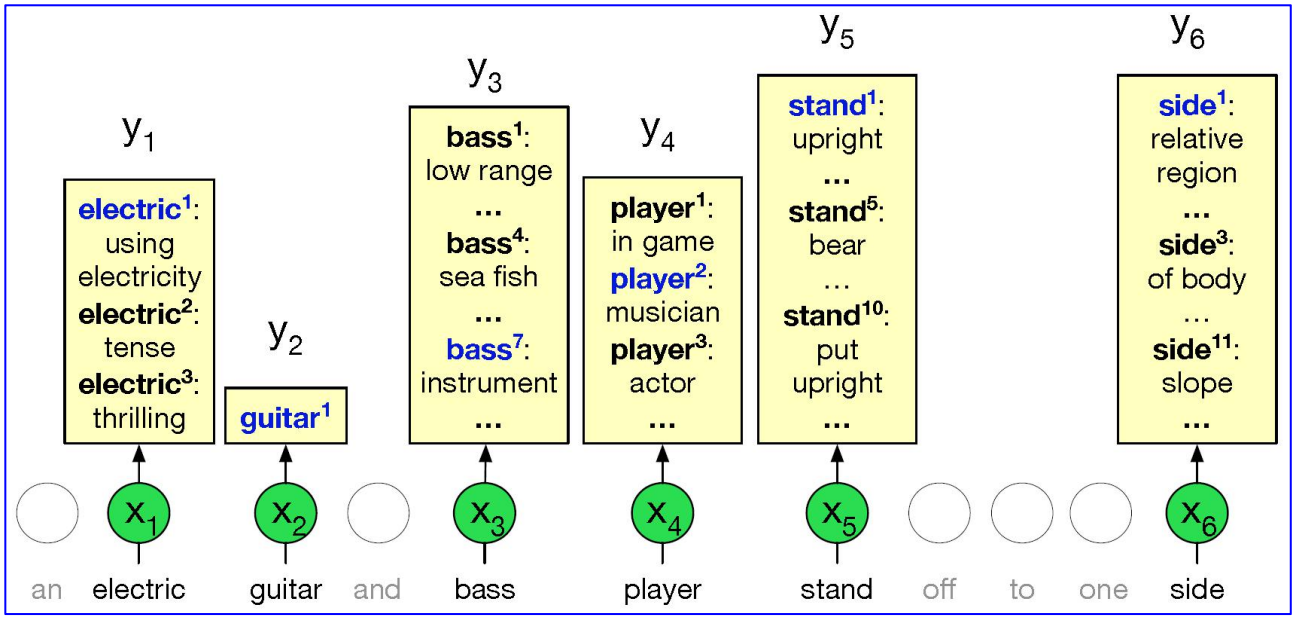

图 18-8：全词 WSD 任务

图注：从输入词(x)映射到 WordNet 意义(y)。只有名词、动词、形容词和副词被映射，注意一些词(如例子中的 guitar)在 WordNet 中只有一种意义。图的灵感来自 Chaplot 和 salakhudinov(2018)。

第二种启发，称为**每一话语一种意义(one sense per discourse)**，是基于 Gale 等人(1992b)的研究，他们注意到在文本或话语中多次出现的单词通常具有相同的意义。这种启发式似乎更适合粗粒度的意义，特别是对于同义词而不是多义词的情况，所以通常不作为基线使用。但是，各种消歧任务通常包括一些偏向于在话语片段内部以相同方式解决歧义的倾向。

18.4.2. WSD 算法:上下文嵌入

表现最好的 WSD 算法是使用上下文词嵌入的简单的 1-最近邻算法，这归功于 Melamud 等人(2016)和 Peters 等人(2018)。在训练时，我们通过任何上下文嵌入(例如，BERT)来传递 SemCore 标签数据集中的每个句子，从而导致 SemCore 中每个标签符记的上下文嵌入。对于每个单词的每个意义 c 的每个符记 c_i，我们对上下文表示进行平均，生成一个 c 的上下文**意义嵌入** v_s:

$$V_S = \frac{1}{n}\sum_i c_i \tag{18.13}$$

在测试时，我们同样计算目标词的上下文嵌入 t，并从训练集中选择其最邻近的意义(与 t 余弦值最高的意义)。图 18.9 说明了这个模型。

我们该如何处理在意义标签的训练数据中没有看到的单词呢?毕竟，SemCor 中出现的意义的数量只是 WordNet 中单词的一小部分。最简单的算法是退回到最常见的意义基线，即在 WordNet 中取第一种意义。但这不是很令人满意。

Loureiro 和 Jorge(2019)提出的一种更强大的方法是，通过使用 WordNet 分类法和超意义，自下而上地估算缺失的意义嵌入。我们得到在 WordNet 分类中的任何更高级别节点的意义嵌入，其方法是，通过对其孩子节点的嵌入进行平均化，因此计算如下三项内容：每个同义词集的嵌入作为其意义嵌入的平均值；一个上位词的嵌入作为其同义词集嵌入的平均值；词汇编目类别(超意义)的嵌入作为该类别的大型同义词集嵌入的平均值。更正式地讲，对于 WordNet $\hat{s} \in W$ 中的每个缺失意义，让其同义词集其他成员的意义嵌入为 $S_{\hat{s}}$，上位词专有的同义词集嵌入为 $H_{\hat{s}}$，而词汇编目的(超意义特有的)同义词集嵌入为 $L_{\hat{s}}$。然后，我们可以如下计算$\hat{s}$的意义嵌入：

$$\text{if} \quad |S_{\hat{s}}| > 0, \quad \mathbf{v}_{\hat{s}} = \frac{1}{|S_{\hat{s}}|} \sum \mathbf{v}_s, \forall \mathbf{v}_s \in S_{\hat{s}} \tag{18.14}$$

$$\text{else if} \quad |H_{\hat{s}}| > 0, \quad \mathbf{v}_{\hat{s}} = \frac{1}{|H_{\hat{s}}|} \sum \mathbf{v}_{syn}, \forall \mathbf{v}_{syn} \in H_{\hat{s}} \tag{18.15}$$

$$\text{else if} \quad |L_{\hat{s}}| > 0, \quad \mathbf{v}_{\hat{s}} = \frac{1}{|L_{\hat{s}}|} \sum \mathbf{v}_{syn}, \forall \mathbf{v}_{syn} \in L_{\hat{s}} \tag{18.16}$$

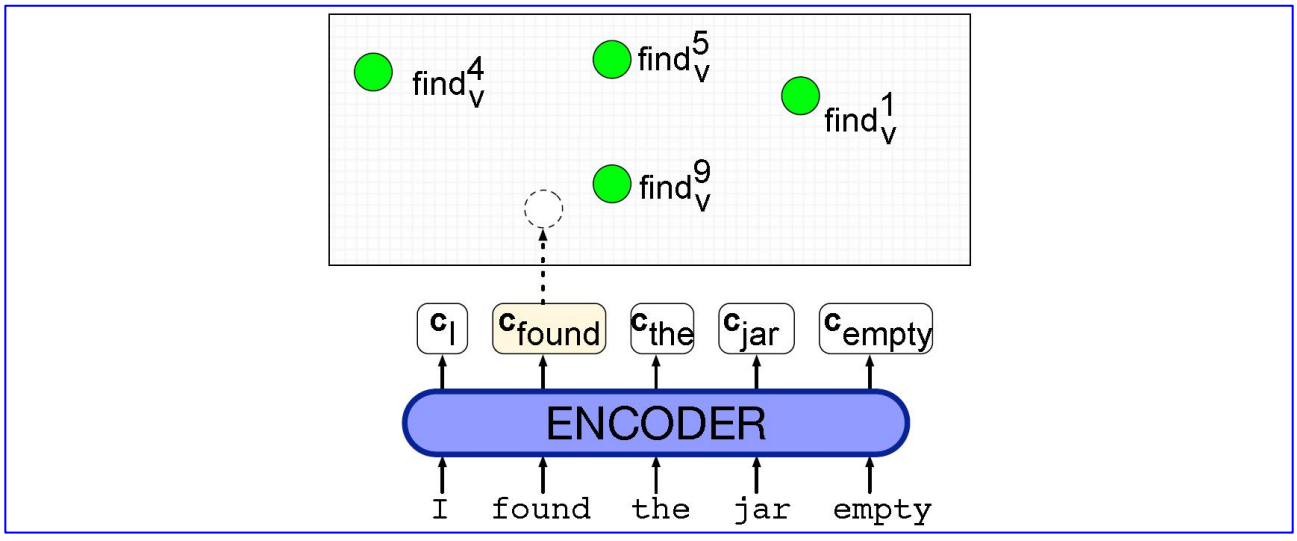

图 18-9：WSD 的最近邻算法

图注：绿色的是预先为每个单词的每种意义计算的上下文嵌入;这里我们只展示了一些 find 的意义。计算目标词 found 的上下文嵌入，然后选择最邻近的意义(在本例中是$find_v^9$)。图的启发来自 Loureiro 和 Jorge(2019)。

由于所有超意义在 SemCor 中都有一些标签数据，所以当算法返回到最一般的(超意义)信息时，算法就可以保证对所有可能的意义都有一些表示，尽管当然是使用一个非常粗糙的模型。

18.5. 可选的 WSD 算法和任务

18.5.1. 基于特征的 WSD

用于 WSD 的基于特征的算法非常简单，功能几乎与上下文语言模型算法一样。表现最好的是 IMS 算法(Zhong 和 Ng，2010)，并通过嵌入进行了增强(Iacobacci 等人 2016，Raganato 等人 2017b)，它使用 SVM 分类器为每个输入单词选择意义，并具有以下简单的周围单词的特征：

•词类标记(用于每边 3 个单词的窗口，在句子边界处停止)

•单词的**搭配(collocation)**特征或长度为 1、2、3 的 n-grams 在每边 3 个单词的窗口内的特定搭配位置(即正好一个单词在右边，或从左边 3 个单词开始的两个单词，以此类推)。

•嵌入的加权平均(所有单词在每边 10 个单词的窗口中，按距离指数加权)

想想《华尔街日报》下面这句话里那个模棱两可的单词 bass 吧:

(18.17) An electric guitar and **bass** player stand off to one side,

如果我们使用一个小的 2 字窗口，一个标准的特征向量可能包括词类、unigram 和 bigram 搭配特征，以及嵌入的加权总和 g，即:

$$[w_{i-2}, \text{POS}_{i-2}, w_{i-1}, \text{POS}_{i-1}, w_{i+1}, \text{POS}_{i+1}, w_{i+2}, \text{POS}_{i+2}, w_{i-2}^{i-1}, w_{i+1}^{i+2}, g(E(w_{i-2}), E(w_{i-1}), E(w_{i+1}), E(w_{i+2})] \tag{18.18}$$

将产生以下向量:

[guitar, NN, and, CC, player, NN, stand, VB, and guitar, player stand, g(E(guitar),E(and),E(player),E(stand))]

18.5.2. 以 Lesk 算法作为 WSD 基线

生成像 SemCor 这样的意义标签语料库是相当困难和昂贵的。另一类 WSD 算法，**基于知识(knowledge-based)**的算法，只依赖于基于 WordNet 知识或其他此类资源，不需要标签数据。虽然监督算法通常工作更好，基于知识的方法可在语言或领域使用词典或字典，但没有意义标签的语料库是可用的。

Lesk 算法是最古老和最强大的基于知识的 WSD 方法，是一个有用的基线。Lesk 实际上是一系列算法，这些算法选择一种意义，此意义的字典注释或定义与目标词的邻域共享最多的词。图 18.10 显示了该算法的最简单版本，通常称为简化的 Lesk 算法(Kilgarriff 和 Rosenzweig, 2000)。

```
function SIMPLIFIED LESK(word, sentence) returns best sense of word
  best-sense ← most frequent sense for word
  max-overlap ← 0
  context ← set of words in sentence
  for each sense in senses of word do
    signature ←set of words in the gloss and examples of sense
    overlap ← COMPUTEOVERLAP(signature, context)
    if overlap > max-overlap then
      max-overlap ← overlap
      best-sense ← sense
    end
return(best-sense)
```

图 18-10：简化的 Lesk 算法

图注：函数 COMPUTEOVERLAP 返回两个集合之间共有的单词个数，忽略虚词或停用列表上的其他单词。最初的 Lesk 算法以一种更复杂的方式定义上下文。

作为 Lesk 算法工作的一个例子，考虑在下面的上下文中消除单词 bank 的歧义:

(18.19) The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

在 WordNet 中有以下两个意义:

$bank^1$	Gloss : Examples:	a financial institution that accepts deposits and channels the money into lending activities “he cashed a check at the bank”, “that bank holds the mortgage on my home”
$bank^2$	Gloss : Examples:	sloping land (especially the slope beside a body of water) “they pulled the canoe up on the bank”, “he sat on the bank of the river and watched the currents”

在(18.19)中， $bank^1$ 的意义有两个与上下文重叠的非停用词:deposits 和 mortgage，而 $bank^2$ 的意义没有重叠词，因此选择 $bank^1$ 的意义。

对简化的 Lesk 有许多明显的扩展，例如通过 IDF(逆文档频率)第 6 章对重叠词进行加权，以减轻像虚词这样的频繁词；最好的表现是使用词嵌入余弦代替词重叠来计算定义和上下文之间的相似度(Basile 等人，2014)。Lesk 的现代神经扩展使用这些定义来计算可以直接使用的意义嵌入，而不是使用 SemCor 训练的嵌入(Kumar 等人 2019, Luo 等人 2018a, Luo 等人 2018b)。

18.5.3. 上下文词的评估

词义消歧是一种比我们在第 6 章中描述的上下文无关的单词相似度任务更精细的单词含义评估。回想一下，像 LexSim-999 这样的任务要求系统匹配人类对两个单词之间上下文无关相似性的判断(cup 和 mug 有多相似?)我们可以将 WSD 看作是一种上下文化的相似任务，因为我们的目标是能够将一个单词含义(比如 bass)与一个上下文(播放音乐)和另一个上下文(钓鱼)中的意义区分开来。

介于两者之间的是**上下文词(word-in-context，WIC)**的任务。在这里，系统给出了两个句子，每个句子都有相同的目标词，但在不同的句子上下文中。系统必须确定目标词在两个句子中是在同一意义上使用还是在不同意义上使用。图 18.11 显示了 Pilehvar 和 Camjo-Collados (2019) WIC 数据集的样本偶对。

WIC 的句子主要来源于 WordNet 中的意义例句。但是 WordNet 的意义是非常精细的。出于这个原因，像 WIC 这样的任务首先将词义聚成更粗的词簇，这样，如果目标词的两个句子上下文在同一个词簇中，就被标记为 T。WIC 将所有意义偶对聚集在一起，条件是：如果它们是 WordNet 语义图中的一级连接(包括姐妹意义)；或者，如果它们属于同一个超意义。在本章的最后，我们指出了其他意义上的聚类算法。

F	There's a lot of trash on the **bed** of the river — I keep a glass of water next to my **bed** when I sleep
F	**Justify** the margins— The end **justifies** the means
T	**Air** pollution— Open a window and let in some **air**
T	The expanded **window** will give us time to catch the thieves — You have a two-hour **window** of clear weather to finish working on the lawn

图 18-11：WIC 数据集的正(T)和负(F)样本偶对

解决 WIC 任务的基线算法使用了像 BERT 这样的上下文嵌入和一个简单的阈值余弦。我们首先计算目标词在两个句子中的上下文嵌入，然后计算它们之间的余弦。如果它超过了 devset 的阈值，我们就会响应为真(这两种意义是相同的)，否则我们就会响应为假。

18.5.4. 维基百科作为训练数据的来源

除 SemCor 以外的数据集已用于全词 WSD。一个重要的方向是使用维基百科作为意义标签数据的来源。当在维基百科的文章中提到一个概念时，文章文本可能包含到这个概念的维基百科页面的显式连接，该页面由唯一标识符命名。这个连接可以用作意义注释。例如，含糊不清的单词 bar 会根据上下文中的意义连接到不同的维基百科文章，包括页面 BAR(Law)、页面 BAR(Music)等等，如下面的维基百科示例所示(Mihalcea, 2007)。

In 1834, Sumner was admitted to the **[[bar (law)|bar]]** at the age of twenty-three,
and entered private practice in Boston.

It is danced in 3/4 time (like most waltzes), with the couple turning approx. 180 degrees
every **[[bar (music)|bar]]**.

Jenga is a popular beer in the **[[bar (establishment)|bar]]**s of Thailand.

然后可以将这些句子添加到受监督系统的训练数据中。然而，为了以这种方式使用维基百科，有必要将维基百科的概念映射到与 WSD 应用程序相关的任何意义清单。从 Wikipedia 映射到 WordNet 的自动算法包括，例如，找到与 Wikipedia 意义在词汇上有最大重叠的 WordNet 意义，方法是经过如下两个步骤：第一步是比较：将 WordNet 同义词集的单词向量、注释及其相关意义与 Wikipedia 页面标题中的单词向量相比较；第二步是传出：传出连接和页面类别(Ponzetto 和 Navigli，2010)。由此产生的映射已被用于创建 BabelNet，它是一个大型的意义标注资源(Navigli 和 Ponzetto, 2012)。

18.6. 使用词典来改善嵌入

词典也被用来改进静态的和上下文的词嵌入。例如，静态词嵌入有反义词的问题。像 expensive 这样的词在将余弦函数嵌入到它的反义词 cheap 中经常非常相似。来自词典的反义信息可以帮助解决这个问题；图 18.12 显示了 GloVE 中目标词的最近邻词，以及此方法之后的一种改进。

Before counterfitting				After counterfitting		
east	west	north	south	eastward	eastern	easterly
expensive	pricey	cheaper	costly	costly	pricy	overpriced
British	American	Australian	Britain	Brits	London	BBC

图 18-12：在 GloVe 中的最近邻居

图注：在 GloVe 中，east，expensive，和 British 的最近邻居包括反义字 west。右侧显示了采用伪造方法后，GloVe 最近邻居的改善(Mrksic 等人，2016)。

解决方案有两个簇。第一个需要再训练:我们修改嵌入训练，以合并词典关系，如同义词、反义词、或超意义。这可以通过修改 word2vec 的静态嵌入损失函数(Yu 和 Dredze 2014, Nguyen 等人 2016)或修改

上下文嵌入训练(Levine 等人 2020, Lauscher 等人 2019)来实现。

第二，对于静态嵌入，更轻量化;在对嵌入进行了训练之后，我们学习了基于词典的第二种映射，这种映射改变了单词的嵌入，使同义词(根据词典)被推得更近，反义词被推得更远。这些方法被称为**改造(retrofitting)**(Faruqui 等人 2015, Lengerich 等人 2018)或**反改造(counterfitting)**(Mrksic 等人，2016)。

18.7. 词义归纳

要建立一个大型的语料库，对每个词的词义进行标签，既昂贵又困难。因此，一种无监督的词义消歧方法，通常称为**词义归纳(word sense induction)**或 WSI，是一个重要的方向。在无监督的方法中，我们不使用人类定义的词义。相反，每个单词的“意义”集合是由训练集中每个单词的实例自动创建的。

大多数用于词义归纳的算法遵循了 Schütze(Schütze 1992b, Schütze 1998)在词嵌入上使用某种聚类的早期工作。在训练中，我们使用三个步骤:

1.对于语料库中单词 w 的每个符记 w_i，计算上下文向量 c。

2.使用聚类算法将这些单词-符记上下文向量 c 聚类为预定义数量的群组或聚类。每个聚类都定义了 w 的意义。

3.计算每个聚类的向量质心。每个向量质心 s_j 是一个**意义向量**，它表示 w 的意义。

由于这是一种无监督算法，我们无法为 w 的每一种“意义”命名;我们只是引用 w 的第 j 个意义。为了消除对 w 的特定符记 t 的歧义，我们同样有三个步骤:

1.计算 t 的上下文向量 c。

2.检索 w 的所有意义向量 s_j。

3.将 t 赋值给最接近 t 的意义向量 s_j 表示的意义。

我们所需要的是一个聚类算法和向量之间的距离度量。聚类是一个经过深入研究的问题，有许多标准算法可以应用于作为数值向量结构的输入(Duda 和 Hart, 1973)。语言应用程序中常用的一种技术称为**凝聚聚类(agglomerative clustering)**。在这种技术中，每 N 个训练实例最初都分配给它自己的聚类。然后，通过连续合并两个最相似的聚类，以自下而上的方式形成新的聚类。这个过程会一直持续，直到达到指定数量的聚类，或者达到聚类中的某种全局良度指标(goodness measure)。如果训练实例的数量使这种方法过于昂贵，那么可以在原始训练集上使用随机抽样来获得类似的结果。

我们如何评估无监督的意义消歧方法？ 像往常一样，最好的方法是在某些端到端系统中进行外部评估。SemEval 评估中使用的一个示例是改善搜索结果的聚类和多样化(Navigli 和 Vannella，2013)。内在评估需要一种将自动派生的意义类别映射到手工标记的黄金标准集合中的方法，以便我们可以将手工标记的测试集合与我们的无监督分类器标记的集合进行比较。例如在 SemEval 任务(Manandhar 等人 2010，Navigli 和 Vannella 2013，Jurgens 和 Klapaftis 2013)中测试了各种此类度量标准，包括聚类重叠度量标准，或通过选择(在某些训练集中)与该类重叠最多的意义将每个意义聚类映射到预定义意义的方法。但是，可以公平地说，此任务的评估指标尚未成为标准。

18.8. 总结

本章涵盖了与词汇术语相关的含义的广泛问题。以下是其中的亮点:

•**词义**是单词含义的所在地;定义和含义关系是被定义在词义层面而不是在词形层面。

•许多词是**一词多义**的，有许多词义。

•意义之间的关系包括**同义关系**、**反义关系**、**整体部分关**系，以及**分类关系**：**下位关**系和**上位关系**。

•**WordNe**t 是一个大型的英语词汇关系数据库，并且 WordNets 支持多种语言。

•**词义消歧(WSD)**是在上下文中确定一个词的正确意义的任务。监督方法使用一个句子的语料库，其中单个单词(词汇样本任务)或所有单词(全词任务)都用 WordNet 这样的资源的意义来做手工标记。SemCor 是最大的带有 WordNet 标签意义的语料库。

•WSD 的标准监督算法是带有上下文嵌入的最近邻居。

•基于特征的算法使用的词类和嵌入词在目标词的上下文中也运行很好。

•WSD 的一个重要基准是 WordNet 中**最常见意义**，即**第一意义**。

•另一个基准是**基于知识**的 WSD 算法，称为 **Lesk 算法**，该算法选择其字典定义与目标单词的邻域共享最多单词的意义。

•**词义归纳**是学习无监督词义的任务。

18.9. 文献和历史说明

词义消歧可以追溯到数字计算机最早的一些应用。Weaver(1955)在机器翻译的背景下首次阐述了作为现代词义消歧算法基础的见解:

> 如果一个人看一本书里的字，一次看一个，就像透过一个有一个字宽的孔的不透明面具，那么显然，一次看一个字是不可能确定这些字的含义的。[...] 但是，如果一个人把不透明掩膜上的缝隙拉长，直到他不仅能看到有问题的中心词，而且还能说出两边的 N 个词，那么，如果 N 足够大，他就可以明确地确定中心词的含义。[...] 实际的问题是:“N 的最小值是多少，至少在可容忍的部分情况下，会导致对中心词含义的正确选择?”

其他的概念在这个早期首先提出包括使用一个同义词典来消除歧义(Masterman, 1957)，监督训练贝叶斯模型来消除歧义(Madhu 和 Lytel, 1965)，以及在词义分析中使用聚类(Sparck Jones, 1986)。

许多消歧工作是在早期面向人工智能的自然语言处理系统的背景下进行的。Quillian(1968)和 Quillian(1969)提出了一种基于图的语言理解方法，其中一个词的定义由一个由句法和语义关系连接的词节点网络来表示，并通过在图中寻找词义之间的最短路径来消歧。Simmons(1973)是另一种有影响的早期语义网络方法。Wilks 用他的优先语义学(Preference Semantics)(Wilks 1975c, Wilks 1975b, Wilks 1975a)提出了最早的非离散模型之一，Small 和 Rieger(1982)、Riesbeck(1975)提出了基于对每个单词丰富的程序信息建模的理解系统。Hirst’s ABSITY 系统(Hirst 和 Charniak 1982, Hirst 1987, Hirst 1988)使用了一种基于语义网络的标记传递技术，代表了这类系统中最先进的系统。与这些主要的符号方法一样，早期的神经网络(当时被称为“连接主义者”)消除词义歧义的方法依赖于带有手工编码表示的小词汇(Cottrell 1985, Kawamoto 1988)。

Kelly 和 Stone(1975)领导了一个团队，为 1790 个有歧义的英语单词手工制作了一套消除歧义的规则，最早实现了一种强有力的经验方法来消除歧义。Lesk(1986)是第一个使用机器可读字典来消除歧义的人。Fellbaum(1998)收集 WordNet 的早期作品。早期使用字典作为词汇资源的作品包括 Amsler’s(1981)使用韦氏大词典和朗曼当代英语大字典(Boguraev 和 Briscoe1989)。

有监督的消歧方法始于 Black(1988)使用决策树。除了 IMS 和基于上下文嵌入的监督 WSD 方法，最近的监督算法包括编码器-解码器模型(Raganato 等人，2017a)。

在监督方法中需要大量的标注文本，这导致了早期对引导方法的研究(Hearst 1991, Yarowsky 1995)。例如 Diab 和 Resnik(2002)的半监督算法就是基于两种语言的平行语料库。例如，法语单词 catastrophe

可能在一种情况下被翻译成英语单词 disaster(灾难)，而在另一种情况下被翻译成英语单词 tragedy(悲剧)，这一事实可以用来消除这两个英语单词的歧义(即选择 disaster 和 tragedy 的意义相近)。

聚类在词义研究中最早使用的是 Sparck Jones (1986);Pedersen 和 Bruce (1997)，Schutze (1997b)，和 Schutze(1998)应用了分布方法。将词义聚类为**粗糙意义(coarse senses)**也已用于解决字典意义过于精细的问题(第 18.5.3 节)(Dolan 1994，Chen 和 Chang 1998，Mihalcea 和 Moldovan 2001，Agirre 和 de Lacalle 2003，Palmer 等人 2004，Navigli 2006，Snow 等人 2007，Pilehvar 等人 2013)。用于训练监督聚类算法的具有聚类词义的语料库包括 Palmer 等人(2006)和 **OntoNotes**(Hovy 等人，2006)。

除其他外，关于一词多义表示的计算方法，参见 Pustejovsky (1995)，Pustejovsky 和 Boguraev (1996)，Martin (1986)，Copestake 和 Briscoe(1995)。Pustejovsky 的**生成词汇**理论，尤其是他的单词的**特质(qualia)结构**理论，是一种在上下文中解释词的动态系统多义性的方式。

WSD 的历史概述包括 Agirre 和 Edmonds(2006)和 Navigli(2009)。

18.10. 练习

18.1 Collect a small corpus of example sentences of varying lengths from any newspaper or magazine. Using WordNet or any standard dictionary, deter mine how many senses there are for each of the open-class words in each sen tence. How many distinct combinations of senses are there for each sentence? How does this number seem to vary with sentence length?

18.2 Using WordNet or a standard reference dictionary, tag each open-class word in your corpus with its correct tag. Was choosing the correct sense always a straightforward task? Report on any difficulties you encountered.

18.3 Using your favorite dictionary, simulate the original Lesk word overlap dis ambiguation algorithm described on page 367 on the phrase *Time flies like an arrow*. Assume that the words are to be disambiguated one at a time, from left to right, and that the results from earlier decisions are used later in the process.

18.4 Build an implementation of your solution to the previous exercise. Using WordNet, implement the original Lesk word overlap disambiguation algo rithm described on page 367 on the phrase *Time flies like an arrow*.