词嵌入的经典方法，六篇论文遍历Word2vec的另类应用

原创 Synced 机器之心 2020-07-14

机器之心分析师网络

作者：王子嘉

编辑：Joni

在本文中，作者首先为读者普及了 word2vec 的基础知识，然后以六篇论文为示例详细介绍了当前研究如何利用经典 word2vec 进行拓展性研究。其中，作者重点介绍的部分是知识嵌入空间的生成过程，对于其完整应用感兴趣的读者可以参阅原论文。

随着深度学习的兴起，每个模型都需要一个输入，而我们现实生活中的对象（文字、图片）等等都不是数字，计算机无法处理。所以如何为每个任务确定一个合适的“输入”就变得尤其重要了，这个过程也被叫做表征学习。

word2vec 做的就是把文字变成对计算机有意义的输入，简单来说就是把这些东西映射到一个空间里，我们平时为了表示位置可能是三维空间，也就是 xyz，但是在图片啊、文本啊这种领域里，三维空间不太够，就可能去到另外一个 N 维空间，在这个空间里，就像三维空间里人的鼻子要跟嘴挨得近一样，我们也希望相似的东西在这个新的空间里也距离近，比如文本里的“鼻子”和“嘴”我们就也希望它们能挨得近一点，因为都属于五官，那么“鼻子”和“腿”就相对离得远一点。

顾名思义，word2vec 是把文字转换成计算机可以识别的输入，所以这个技术最开始应用、也是应用最多的地方就是自然语言处理领域（NLP）。其实在之前对于表征学习，我基于 ICLR 和 CVPR 做过两次 high-level 的总结，但是这次这篇文章主要着眼于 word2vec，从细节着手，看看 word2vec 中发现的空间是如何被**改进**并**使用**的，同时也看一下基于 word2vec 的原理发现的**新空间**。在开始正题之前，为了防止有人不清楚 word2vec 从而影响对后文的理解，这里科普一下本文会用到的相关基本概念。

1、word2vec 简介

什么是 word2vec： Word2Vec 是一个过程（技术），在这个过程中，将文本作为神经网络的训练数据，这个神经网络的输出向量被称作嵌入，这些嵌入（向量）在训练后会包含单词的语义信息。这个过程做的就是从每个单词有多个维度的空间嵌入到具有低得多维度的连续向量空间。向量

嵌入可以在真实对象的“离散”世界和机器学习的“可微分”世界之间架起一座桥梁，因此在数据库研究方面具有巨大的潜力。一类对象的嵌入向量是从 X 映射到某个向量空间，称为潜空间，我们通常假设它是一个有限维 d 的实向量空间 R^d。

它们用在哪里：最终 word2vec 就会产生如图 1 所示的一堆向量（word embedding，词嵌入），这些向量就可以作为后续任务中神经网络模型的输入。

图 1：word embedding 示例。图源：https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/

为什么要用 word2vec：如前文所述，这些嵌入抓住了文本的语义，相似含义的词会具有更近的距离（图 2 展示了其中一种、也是最常见的相似度衡量方式——余弦相似度）。而且经过长久的实践，研究者都发现这种语义的编码使得各种 NLP 任务都表现得很好。

图 2：衡量距离的距离示例——cosine similarity。图源：https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

基本模型：训练 word2vec 的常用方法有 CBOW 和 skip-gram。如图三所示，w(t)表示当前单词，w(t-?)表示前面的单词，w(t+?)表示后面的单词，简单来说，CBOW 就是使用周围的单词来预测当前单词，而 skip-gram 模型利用当前单词尝试预测周围大小为 c 的窗口中的单词。具体计算细节可以看一下图 3 的来源那篇文章，这里就不做详细介绍了。

图 3：skip-gram 和 CBOW。图源：https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314

如何去衡量：在使用嵌入的时候，我们应该考虑几个问题——对象的哪些属性用嵌入表示？我们用的距离测量的意义是什么？潜空间的几何性质是否对应了 X 上有意义的关系？当然同时，我们也要考虑计算向量嵌入的计算成本。

2、完善旧空间

了解了 word2vec 的基本常识，就可以正式进入正题了。在上一节说过，我们在建立一个嵌入空间的时候，我们要考虑的是对象的那些属性需要在嵌入空间中被表征。word2vec 在传统的 NLP 任务中表现得很好，但是在一些新的、较为复杂的任务中，有一些属性就不能很好的被体现了，因为最开始 word2vec 模型是完全基于文本进行训练，而很多关系是在文本中很难体现出的，比如“看”和“吃”，单独看这两个词，就算是我们也很难想到它们是有联系的。

但是如果加上图 4，它们是不是就联系起来了，这张图的描述可以是两种，一种是小姑娘正在“看”冰激凌，另一种则加入了一定的联想——小姑娘正在“吃”冰激凌，在这张图的描述中，这两句话都是对的，这个例子除了解释了利用纯文本进行学习的缺陷，也侧面说明了这种图片描述之类的任务中，这种信息也是很重要的。

图 4：吃冰激凌的小姑娘。图源：[1]

这也不难理解，人类在感知这个世界的时候用到的模式不只是看，还有听、闻等等，同样的，在看的时候，看的也不只是书面上的文字，而语言却是传递知识的载体，所以现在研究者**开始用不同的感知模式（声音、图片）来学习语言模型（multi-modal learning）**，从而让语言模型学出的嵌入能够更加全面的表征我们人类的理解能力。

至于技术层面，其实回溯到语言模型最开始起源的那阶段，语言模型跟迁移学习类似于一对兄弟，只不过进入了不同的领域，也就有了不同的名字。比如说在 NLP 任务中，语言模型先被训练好，然后后面直接用预训练好的语言模型来进行下面的任务，像不像冻住了前面几层的迁移学习（如果这个看不懂，也不太影响后面的理解，觉得放不下的可以看看机器之心之前的文章，有很多基础教程的，这里就不做介绍啦）。

之所以说迁移学习，是因为如果想要达到完善 word2vec 空间的效果，其实就是类似于迁移学习里的全局 finetune，把前面的语言模型（一开始冻住的层）也放进训练里来。

为了展示当前研究具体是怎么利用 word2vec 进行拓展应用，这一节对四篇论文进行简要介绍，从而展示图片和声音是如何加强已有的 word2vec 嵌入空间的表征能力的。当然这里介绍的知识嵌入空间的生成过程，原论文中还有其他的创新点，如果有兴趣可以再去拜读一下完整论文。

2.1 Visual Word2Vec (vis-w2v) [1]

论文链接：https://arxiv.org/pdf/1511.07067.pdf

这个方法解决的就是图 4 提到的例子里的问题，这里作者要在原有词嵌入的基础上加入对图片语义的表征，也就是将图片也作为背景加入到 w2v 的训练中去。**这个方法基于 CBOW，并用到了辅助标签（surrogate label）作为图片背景的表征。**

图 5：vis-w2v 中的网络结构。图源：[1]

具体模型如图 5 所示，模型的输入是图片 - 文字对 D = {(v, w)}，v 是指图片的特征，w 是 v 对应文本描述，这里 w 的形式（句子或词）会根据场景不同而改变（下面会详述）。

图 5 中展示的是一个窗口（这个窗口可能包含 w 的部分词，也可能包含完整的 w，也是根据场景不同而改变，下面也会详述），这里的的 w_1 到 w_l 表示一个窗口中包含的词，由 one-hot 方式进行编码（N_V 表示 one-hot 编码的维度），H_wi 是由 wi 乘 W_I 获得的，这里的 W_I 是共享的，就是所有的 w_i 乘的都是同一个 W_I，因为是 one-hot 编码，其实也就相当于取了 W_I 的某一行，最终的 H 是由各个 H_wi 取平均获得的：

获取了 H（N_H 就是嵌入的维度）之后，这个 H 就作为最终的特征向量，利用 W_O 将其进行映射 N_K 维（一共有 N_K 类，这里的 N_K 后面解释），对其进行 softmax 操作之后，就可以知道这段文本（w_1-w_l）属于哪一类了，这一步就是做了一个分类。

如果了解 NLP 任务的话，H 的获取的第一步跟平时我们获取 embedding 的方式一样，因为这里的 W_I 是初始化为传统 CBOW 的权重，所以这一步其实就是获取这些词的传统嵌入，然后对其取平均作为整个窗口所有词的特征，然后做了一个分类任务。

现在就是这篇文章的核心了——上一段说了这是个分类任务，那么分类任务的 label 从哪来的呢？这就是辅助标签的作用了——**作者将图片 v 进行聚类，聚类成 N_K 类，然后每个 v 所属的类就是这个 v 对应的 w 在做分类任务时候的 label。**

然后回到 w，这里的 w 允许对各种形式的 w 进行选择，比如完整的句子或形式的元组 (主要对象、关系、次要对象)。w 的选择取决于我们所关心的任务。例如，在常识断言分类和基于文本的图像检索中，w 是元组中的短语，而在视觉释义中，w 是句子。给定 w，S_w 也是可调的。它可以包括所有的 w(例如，从元组中的一个短语学习时) 或单词的一个子集(例如，从句子中的一个 n 元上下文窗口学习时)。

最后再说一下效果，这个任务其实就是在 w2v 的基础上进行 finetune，这种直接进行全局 finetune 的效果，据论文中所说，也可以保持最开始 w2v 的效果，就是如果一些词在 finetune 过程中没有出现，它们就会保持原有的语义特性，这种全局 finetune 并不会让新型的 w2v 在传统任务中变得更差。

如下表 1 所示的视觉转述任务对比中，vis-w2v 的效果比单纯的 w2v 任务要好很多。

表 1：转述任务平均准确率（AP）。表源：[1]

2.2 Visually Supervised Word2Vec (VS-Word2Vec) [2]

论文链接：https://ieeexplore.ieee.org/abstract/document/8675640?casa_token=lw7U2LlGUk4AAAAA:uhM9BVykvRQyYoWE5KCq3BfjUSjLRED2yV7nktCUgw3jDcAh_R2xx8iV7Az3pBWTZPBQ87cQzEgd

这篇论文也是想将图片中的信息传递给 w2v，让 w2v 空间能够更好地表征图片中的信息。上一篇论文是将广义的图片信息加入到 w2v 中（通过图片的相似度来引导词的相似度），而这篇论文的全称则是 Embedded Representation of Relation Words with Visual Supervision，顾名思义，就是为了让嵌入空间能够更好地表征关系词（“我拿着包”里的 “拿着” 就是关系词）。

下图 6 中展示了一些关系词的例子，[3]则给出了类似于这样的一个数据集，这是本篇论文的输入形式之一。

图 6：关系词。图源：[3]

VS-Word2Vec 的基本结构如图 7 所示，这个架构分为上下两部分。上面是一个 CNN，类似于图 6 中的图片是输入，输出是一个特征向量，这个特征向量就作为这个图片对应的关系词的嵌入（Visual relation feature space）了；下面是传统的 CBOW，也会产生词嵌入。

本文除了与第一篇论文要表征的信息不同之外，所用的方法也不一样，其**根本思想就是：如果这个词是关系词，那么就让上面（CNN）产生的词嵌入跟下面产生的嵌入尽量相同，但是如果这个词不是关系词，那就不管上面的，跟传统的 CBOW 训练完全一样。**

图 7：VS-Word2Vec 网络结构。图源：[2]

具体来说，整体算法如图 8 所示，第一二行对应的是图 7 上半部分的 CNN，先计算图表征（Visual relation feature，算法第 4 行）：

这里的 y_wi 就是某个关系词 wi 的表征，具体就是利用 VGG 来得到这个关系词对应的所有图片（Q_i 张图）的特征向量（y_wi^q），然后对这些特征向量取平均值。前面说过，如果训练的词属于关系词，那么作者希望通过 CBOW 和通过 CNN 生成的两种 embedding 尽量一致，本文中的不一致性则由下式衡量：

这里的 s_ij 表示关系词 i 和关系词 j 的余弦相似度，右下角标的 r 代表这是关系词，x 则代表 CBOW 中产生的词嵌入，v 则代表 CNN 中生成的词嵌入，这个式子中 J 越小越好。

图 8：VS-Word2Vec 算法流程。图源：[2]

最后就是图 8 中的第 4 行到第 19 行了，这就是上述整体思想的体现，也就是在计算下式，同时进行参数的更新（梯度上升）：

首先 D 是 CBOW 的训练集，然后对于这个训练集中的每个词，计算两个损失 e1 和 e2，e1 就是前半部分，也就是传统 CBOW 的损失，e2 是后半部分，衡量了两种 embedding 的不一致性（J_V），\lambda 就是个可调的超参数，p_t 表示 w_t 是不是关系词，如果是关系词，那么 p_t=1，保留后半部分，如果不是关系词，则变成传统的 CBOW 训练过程。

注意看第 15 行，这里是用了一个 for loop，所以本文的目标并不是让上下两部分对一个关系词产生完全相同的嵌入，而是要求关系词能够保证图 7 中上下的"一致性"，所以用的是 J_V，而不是直接使用余弦相似度。

使用这种方法后，作者在 SimVerb-3500 中的九个类别的同义词（SYNONYMS, ANTONYMS, HYPER/HYPONYMS, COHYPONYM，NONE ）进行了对比，对比结果如表 2 所示，总体来说是比 CBOW 要好的，而且一些特别的类中，效果的提升还很大。

表 2：同义词一致性结果。表源：[2]

2.3 Action2Vec [4]

论文链接：https://arxiv.org/pdf/1901.00484.pdf

又一篇对视觉信息进行探索的文章，不过这里探索的对象变成了视频（因为是对动作进行编码），如图 9 所示，思想跟上文差不多，不过变成了左右结构，而且这次是实实在在的两词比较（pairwise ranking loss），而非利用一致性进行判断。

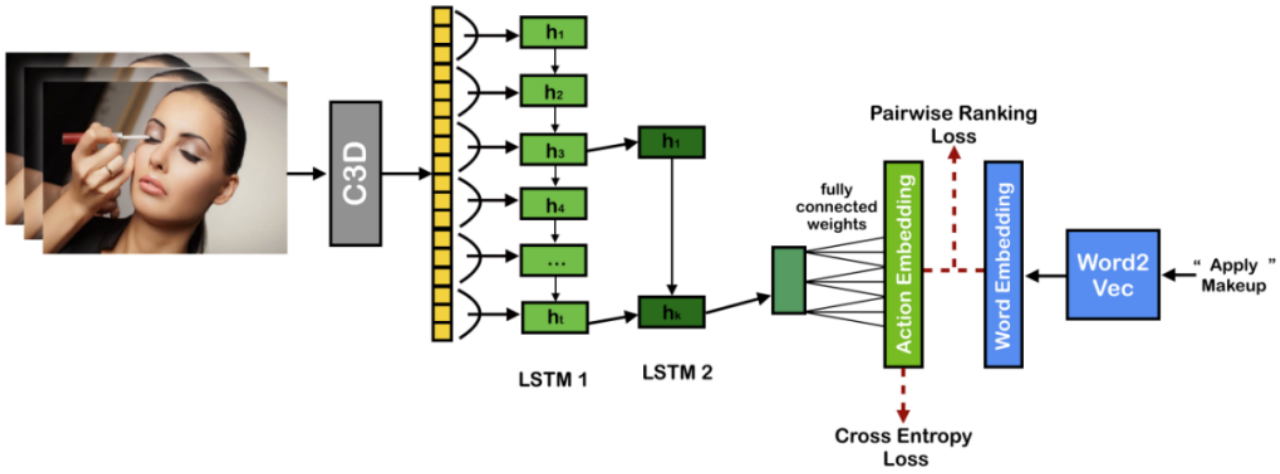

图 9：Action2Vec end-to-end architecture。图源：[4]

具体来说，左边先使用由 [5] 中数据集预训练好的 C3D 模型来提取每一帧的图片的特征向量，然后使用了一个 Hierarchical Recurrent Neural Network (HRNN)，并加入了自注意力机制，最后通过一个全连接层（fully connected weights）将 LSTM2 得到的视频嵌入变成跟词嵌入相同维度的向量，最后这个向量又通过一个全连接层做了一个分类任务，判断这个视频对应的动作是什么。然后通过一个“双损失”（cross entropy+pairwise ranking）来让改善后的 joint embedding space 同时具有视频和文本的语义信息。

HRNN：这里的 HRNN 就是指用两层 LSTM，第一层用于提取局部特征（输入是每一帧的图片），第二层 LSTM（LSTM2）的输入则是每隔 s 张图片的 LSTM1 的输出，以图 9 为例，这里的步长为 3，每三张图片（h1-h3,h4-h6,...）会输出一个向量，这些向量就是 LSTM2 的输出了。自注意力机制里面的注意力计算这里就不做介绍啦，

双损失（dual loss）：这里的双损失就是指 cross entropy 加 pairwise ranking loss，cross entropy 这里就不多做介绍，对应的就是上面说的分类任务，pairwise ranking loss(PR loss)如果不了解的可以看一下这篇文章（https://gombru.github.io/2019/04/03/ranking_loss/），会更容易理解下面的式子。这里的 PR loss 定义为下式：

$$\mathcal{L}_{PR} = \min_{\theta} \sum_{i} \sum_{x} (1 - s(a_i, v_i)) +$$
$$max\{0, s(a_x, v_i)\} + max\{0, s(a_i, v_x)\}$$

$$\mathcal{L}_{Dual} = \mathcal{L}_{PR} + \lambda * \mathcal{L}_{CE}$$

这里的 a_i 和 v_i 分别表示 HRNN 和 word2vec 模型产生的动作词 i 嵌入，a_x 和 v_x 则分别对应 HRNN 和 word2vec 产生的负样本（也就是非动作词 i 的嵌入）。这里注意了，图 9 画的并不全，cross entropy（CE）对应的分类任务并没有画在上面，action embedding 先通过一个全连接层做分类任务，然后才有的 CE 损失。

最后需要说明的一点是因为两个数据库的词并不能完全一致，可能会出现视频数据库中的词在 word2vec 词库中不存在的情况，这时这些动词就会被转化成对应的形式（如 walking 变成 walk 等）。

实验部分，作者在 ZSAL(Zero Shot Action Learning)任务中与其他 ZSL 模型进行了对比，可见作者提出的模型在各个数据集上的效果都是最好的。

Train-Test Split	HMDB51	UCF101	Kinetics
50/50			
Action2Vec w/dual loss + attention	**23.48**	**22.10**	**17.64**
Pooled C3D fc7	5.00	11.41	9.89
TZS w/out aux data + w/ knowledge test labels [35]	19.10	20.8	-
TZS w/out aux data + w/out knowledge test labels [35]	14.50	11.70	-
SAV w/out aux data [36]	15.00	15.80	-
UDA [17]	-	14.00	-
80/20			
Action2Vec w/dual loss + attention	**40.11**	**36.51**	**22.93**
KDCIA [9]	-	31.1	-
KDCIA [9]	-	29.6	-
UDA [17]	-	22.50	-

表 3：ZSAL(Zero Shot Action Learning)模型效果对比。表源：[4]

2.4 sound-word2vec [7]

Sound-Word2Vec: Learning Word Representations Grounded in Sounds

Ashwin Vijayakumar[1], Ramakrishna Vedantam[2] and Devi Parikh[3,1]
[1] Georgia Tech, [2] Virgina Tech, [3] Facebook AI Research
{ashwinkv,parikh}@gatech.edu, vrama1@vt.edu

论文链接：https://arxiv.org/pdf/1703.01720.pdf

前几篇介绍了视觉，最后一篇论文我们开始涉及到了听觉，也就是声音信号。虽然大部分声音都存在拟声词，但是很多拟声词在文本中并不常见，而且相比于直接的声音，这些词对应的语义信息很难被学习到，因此本文作者将这些叫声的声学特征整合到了传统的词嵌入空间中。值得注意的是，这里又用到了辅助标签（聚类）。这个模型的整体结构如图 10 所示.

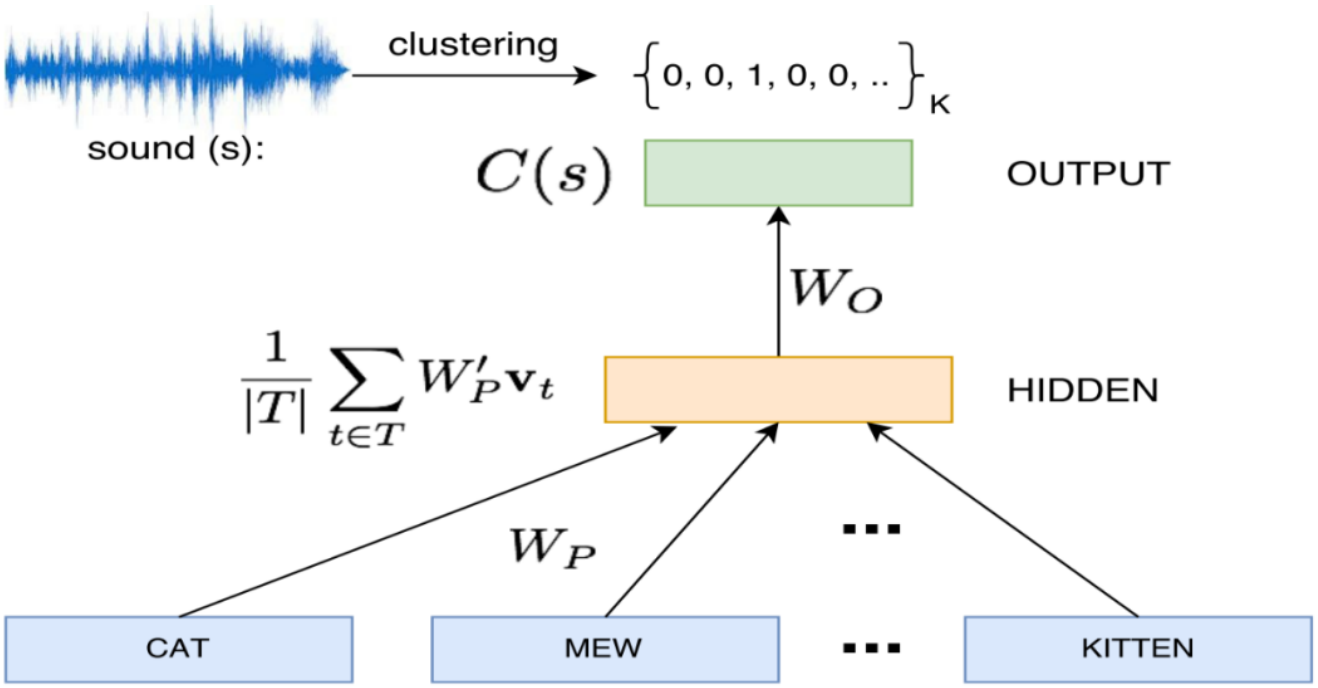

图 10：sound-word2vec。图源：[7]

图 10 中，这个模型的输入是基于 FreeSound 生成的输入对{s, T}，s 是声音，T 是用户提供的这个声音对应的一系列 tag（一组词），这些 tag 首先通过 W_P（由预训练好的 word2vec 模型中的 weights 来初始化）变成嵌入，然后对这些嵌入进行取均值得到隐藏层的值 H，最终 H 通过全连接层（W_O）完成一个分类任务，输出一个类别。

与第一篇论文一样，这里分类任务的标签又是来自于聚类。s 先经过聚类得到类别标签，然后这个标签就用于训练 W_P 和 W_O。整体上来说，这篇论文的思想跟第一篇论文差不多，但是论文中关于声音如何表征等的声音处理还是很具有启发性的，同时这篇论文再一次证明了整体 finetune 可以在改善传统词嵌入空间上取得不错的效果。

这个方法的效果如表 4 所示，在普通词上，效果跟 word2vec 差不多，但是在拟声词上，sound-word2vec 明显表现就要好很多了。作者还在一些基于文本的拟声词识别任务中进行了实验，效果比普通的 baseline 模型都要好很多（具体细节可以去看一下原论文）。

word	word2vec	sound-word2vec
apple	apples, pear, fruit berry, pears, strawberry	bite, snack, chips chew, munch, carton
wood	lumber, timber, softwoods, hardwoods, cedar, birch	wooden, snap, knock, smack, whack, snapping
bones	skull, femur, skeletons, thighbone, pelvis, molar	eggshell, carrot, arm blood, polystyrene, crunch
glass	hand-blown, glassware, tumbler, Plexiglass, wine-glass, bottle	shattered, ceramic, smash clink, beer, spoon
Onomatopoeic query words		
boom	booms, booming, bubble, craze, downturn, upswing	bomb, bang, explosion bombing, exploding, ecstatic
jingle	song, commercial, catchy-tune, ditty, slogan, anthem	magic, tinkle, nails bells, key, doorbell
slam	slams, piledriver, uranage spinkick, hiptoss, hit	shut, lock, opening closing, latch, door
quack	charlatan, quackery, crackpot homeopaths, concoctions, snake-oil	duck, snort, calling chirp, tweet, oink

表 4：相似词示例。表源：[7]

3、探索新空间

From context to concept: exploring semantic relationships in music with word2vec

Ching-Hua Chuan[1] · Kat Agres[2] · Dorien Herremans[2,3]

论文链接：https://link.springer.com/article/10.1007/s00521-018-3923-1

Google 在去年利用语言模型将蛋白质序列转换成嵌入，从而实现了很多相关任务的飞跃，这个我曾经写过一篇文章来专门介绍，这里就不多做赘述，今天主要介绍这个方法是如何应用于音乐上，从而产生一个新的基于音乐的嵌入空间，新的嵌入空间乐理知识进行了表征——**音乐 + word2vec [6]**。

因为这篇论文包含了比较专业的乐理知识，而对应的机器学习方法就相对比较传统，就是一个 skip-gram 模型加上对乐谱进行编码。但是论文中对乐谱各项乐理知识在乐谱嵌入空间的表征情况进行了详细分析，表明了 skip-gram 很好地从乐谱中学习到了乐理知识。

关于乐谱的编码，如下图 11 所示，图中包含了肖邦: 玛祖卡舞曲 编号 67 第 4 首的前六个小节（Chopin's Mazurka Op. 67 No. 4 ），以及前三小节的编码实例。这里相当于把一拍当做文字中 j 的一个字，第一块包含 E，即四分音符中表示音高 E5 的音高类。由于第二拍的音高为 E5 和 A3，所以第二块包含了 E 和 A。注意，作者在第二小节中包含了 E，尽管音高 E5 与第一拍是连在一起的 (不是一个开始)，但它仍然在第二拍中发出。同样，由于第三个节拍包含音高 E3, A3, E4, E5(来自于点连音) 和 F5，所以第三块包含音高类 E, A, f。图 2 中的例子也可以用来解释选择节拍作为切片持续时间。

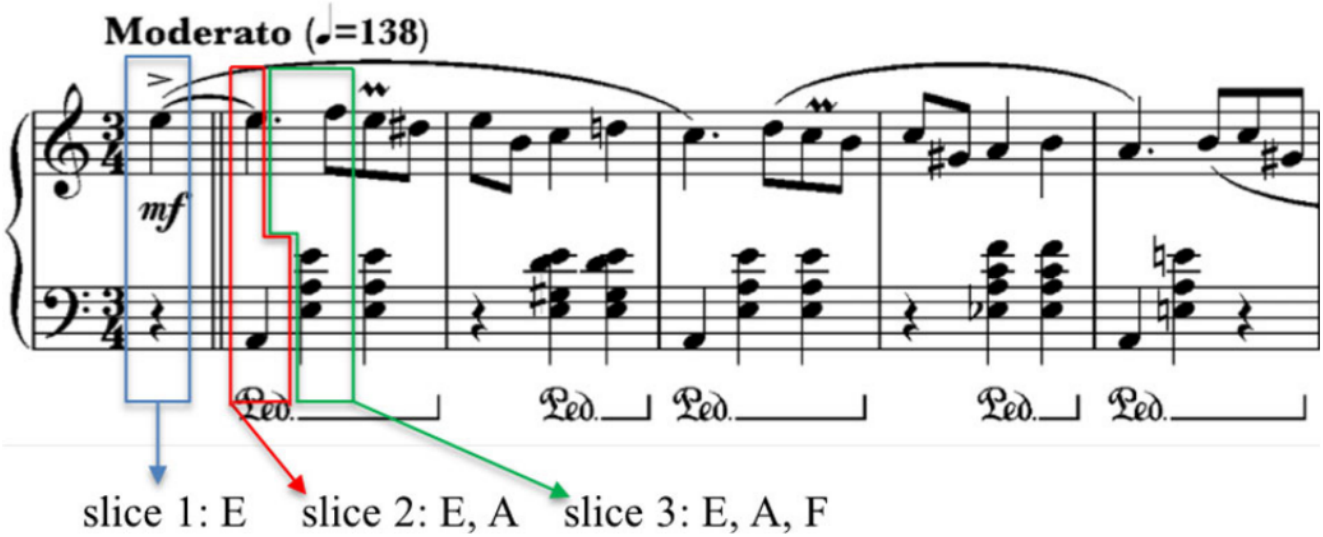

图11：乐谱分类。图源：[6]

如果音片长于一拍，我们可能会失去音调和和弦变化上的细微差别。相反，如果切片短于一个节拍，则可能有太多重复的切片(其中切片之间的内容是相同的)。寻找切片的最佳持续时间也很重要，但是这篇文章并没有涉及，相信一个更好的编码方式会让这个研究的效果更好。

因为这篇文章的价值不在于用了什么机器学习方法，只是用了 skip-gram，故而不再对训练过程进行讲述啦。当然，这篇论文除了证明了 skip-gram 可以在音乐领域很好的获取 chord 和 harmonic 特征，还提供了很多音乐领域可以用的数据集（section 4），如果有兴趣在这个领域做点什么，这些数据集还是很有用的。

这篇论文的结果分析过于专业，如果想看一下音乐大师对这个模型的评价，可以去看一下原始论文，总之这个模型在各个方面表现的很好，对音乐有了解的同学可以看一下原论文中是怎么分析的，或许对后续任务也有很大的帮助。

4、使用这个空间

Word2vec to behavior: morphology facilitates the grounding of language in machines.

David Matthews University of Vermont
Sam Kriegman University of Vermont
Collin Cappelle University of Vermont
Josh Bongard University of Vermont

论文链接：https://arxiv.org/pdf/1908.01211.pdf

最后，在讨论了如何改善传统词嵌入空间和如何创建新嵌入空间之后，如何使用这个空间也很重要。但是因为本文不是对 word2vec 的介绍，所以传统的 NLP 任务中 word2vec 的应用在此就不再多做介绍了，网上已经有很多实践上或是理论上的科普文。这里主要介绍词嵌入是如何在 RL 中应用的——**Word2vec to behavior [8]**。

这里的 a 代表声学神经细胞，一开始先输入命令，然后用这个命令的 embedding 初始化隐藏层 hi，这里初始化是使用文中 5 个命令词的 embedding 来先进行预训练以初始化 h_1-h_5，命令词为‘forward’, ‘backward’, ‘stop’, ‘cease’, ‘suspend’, and ‘halt’，其中后面四个词表达的意思一致，所有一个不会被用来做初始化，作为测试组。初始化完成后，这些虚线的连接就会被删除，然后机器人就进入仿真器开始仿真，将命令的嵌入输入给机器人，然后通过各个传感器 (s) 得到的信息进行动作。这个初始化就使得网络获取了语义信息。

除了上述机制，图 12 中的整个网络并不复杂，第一层叫传感器层，从机器人的传感器获取数据，然后这些神经元跟后面的隐藏层进行全连接，这里的第二层隐藏层是一个带自连接的 recurrent neural network，最后这个隐藏层又跟最后的动作层（最右边）进行全连接。

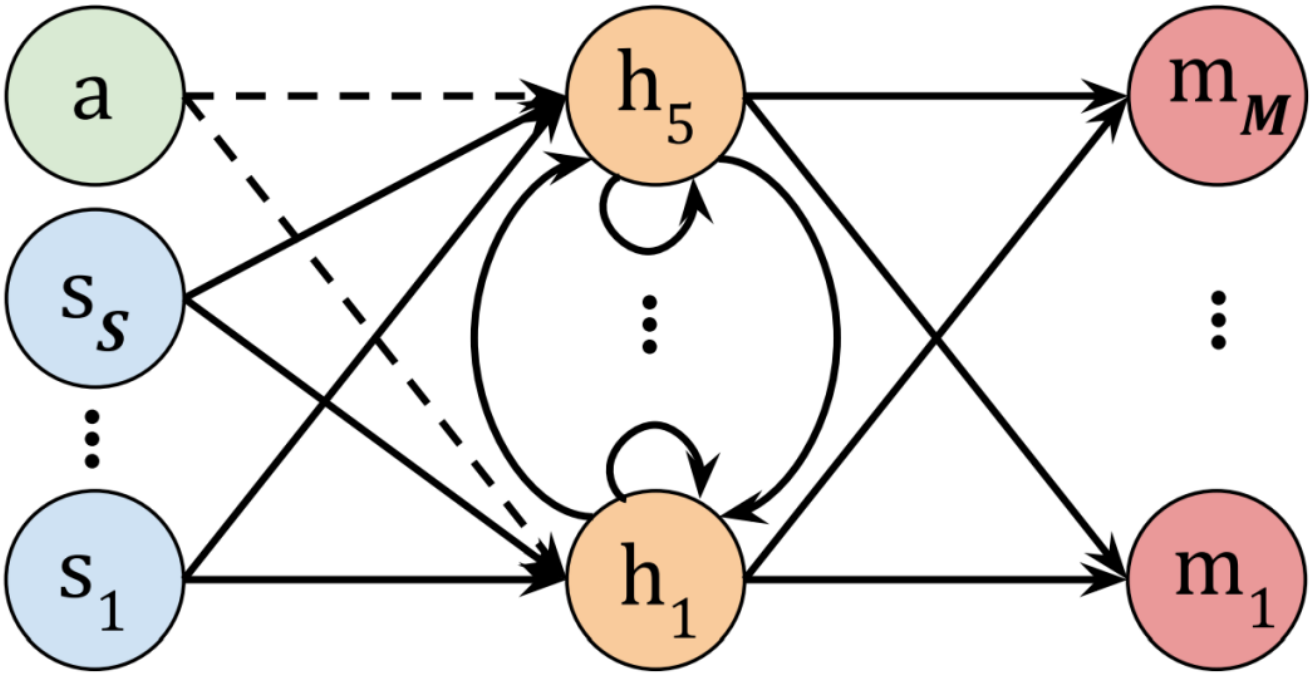

图 12：Word2vec to behavior 训练网络。图源：[8]

总结起来，整个训练流程如图 13 所示，通过向输入层中的神经元 a 提供与“停止”等命令相关的 word2vec 嵌入，可以设置机器人控制策略的隐藏层的初始值。然后将该策略下载到机器人上，并将通过其运动生成的传感器数据提供给输入层的其余部分（虚线箭头），从而进一步更改隐藏层和电机层。

经过评估后，将根据与命令配对的目标函数（例如惩罚运动的函数）对机器人的行为进行评分。然后，针对其他四个命令和目标函数，对同一策略进行四次以上的评估（B 和 C 分别对应两次），对策略进行训练，以针对所有这五个功能（D）最大化平均分数。经过训练后，最佳策略会提供一个训练时没有的第六种同义词“cease”，并且其行为会根据“停止”目标函数（E）进行评分。

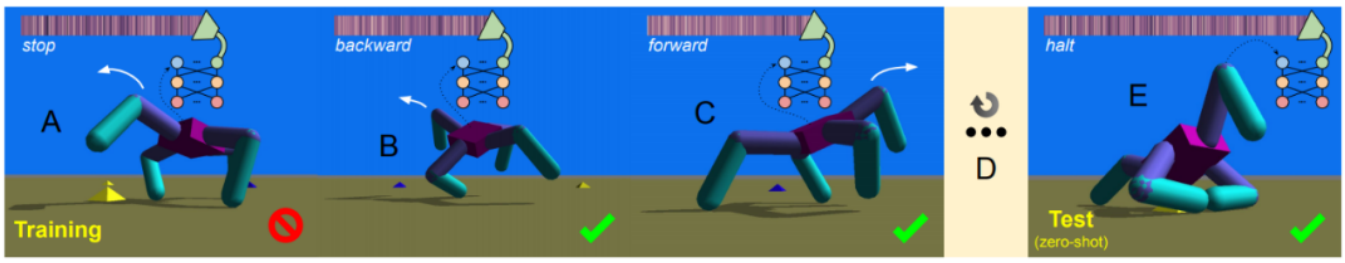

图 13：训练流程。图源：[8]

最终的结果如下图所示，每个颜色代表一种命令，可以看到作者的方法（第一个）训练的机器人在“停止”这条命令上确实表现得比其他的要好。

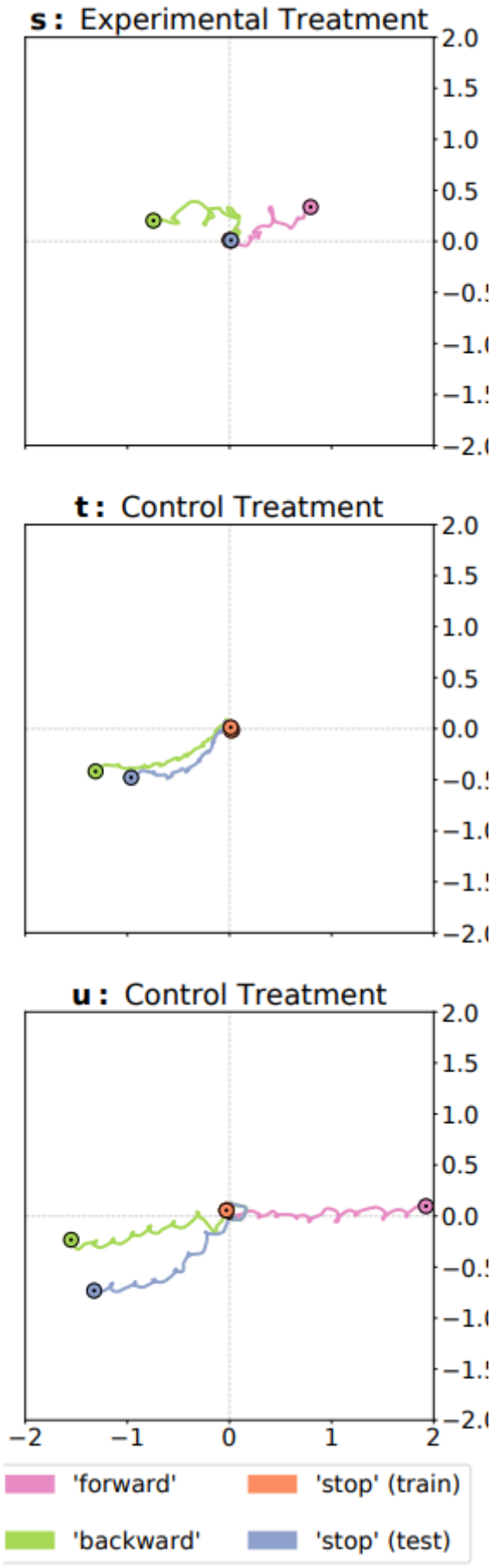

图 14：实验结果。图源：[8]

总结

从本文提到的这些分析性或是创新性论文来看，skip-gram 和 CBOW 能够很好地获取我们生活生很多对象的语义（音乐、声音等），而 multi-modal 是一个很好地完善现有嵌入空间的方法，在没有 label 的情况下，合理的聚类也可以提供给模型辅助标签。这个嵌入空间也不只是可以应用于 NLP 领域，还有很多其他领域可以直接套用 w2v 中生成的嵌入空间（如 RL）。

当然，未来还有很多其他可以探索的方向，比如发展比较初级的音乐领域，如何将声音中的情绪结合到传统的 w2v 模型中去等等。

总之，语言作为我们观察和描述世界的一个基本要素，语言 基本覆盖了我们生活的方方面面，在某些层面也反映了客观世界的规律（比如语言学的"复合性原理"-compositionality)，遇到无法解决的学习的问题的时候，语言模型或许会给你一点点启发。

参考文献

[1] Kottur, Satwik, et al. "Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[2] Wang, Xue, et al. "Embedded Representation of Relation Words with Visual Supervision." 2019 Third IEEE International Conference on Robotic Computing (IRC). IEEE, 2019.

[3] Lu, Cewu, et al. "Visual relationship detection with language priors." European conference on computer vision. Springer, Cham, 2016.

[4] Hahn, Meera, Andrew Silva, and James M. Rehg. "Action2vec: A crossmodal embedding approach to action learning." arXiv preprint arXiv:1901.00484 (2019).

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In CVPR, 2014. 4

[6] Chuan, C.-H., Agres, K., & Herremans, D. (2018). From context to concept: exploring semantic relationships in music with word2vec. Neural Computing and Applications. doi:10.1007/s00521-018-3923-1

[7] Vijayakumar, Ashwin K., Ramakrishna Vedantam, and Devi Parikh. "Sound-word2vec: Learning word representations grounded in sounds." arXiv preprint arXiv:1703.01720 (2017).

[8] Matthews, David, et al. "Word2vec to behavior: morphology facilitates the grounding of language in machines." arXiv preprint arXiv:1908.01211 (2019).