中图分类号: TP183 学校代码: 10055
UDC: 004.8 密级: 公开

南開大學

硕 士 学 位 论 文

利用基于层次注意力机制的变分自编码器进行韵律文本学习

Rhythmic Text Learning with Hierarchical Attention Mechanism Based Variational Autoencoder

论文作者 王灏正 指导教师 杨征路

申请学位 硕士 培养单位 计算机与控制工程学院

学科专业 计算机科学与技术 研究方向 数据挖掘

答辩委员会主席 评 阅 人

南开大学研究生院

二〇一八年四月

摘要

韵律文本是一种特殊类型的文本，相对于通俗的自然语言，韵律文本分析与学习值得进行专门的研究。常见的韵律文本主要包括诗歌、歌词和一些其他艺术形式中的唱词。与一般的自然语言不同，韵律文本通常是有一定音韵特征的。韵律文本的韵律是其主要特征，在进行韵律文本学习时，除了其语义特征，韵律特征也是需要着重关注的一点。

现有韵律学习方法大多需要人为规定一些韵律特征来进行学习，甚至完全忽略韵律文本的韵律特征，仅仅考虑其语义特征及其形式。

为了解决以上问题，本文提出了一个新方法，即 rhyme2vec，来学习韵律表征向量。这个方法包含两个模型，即连续行韵律和隔行韵律。通过整合这两个模型，rhyme2vec可以很好地处理韵律模式的多种特征。

本文还提出了一个结合了层次注意力机制的变分自编码器的框架用于融合韵律文本的韵律特征和语义特征。该框架旨在处理韵律文本的表征学习问题，整合了多种未被探索的机制，即，利用注意力机制对韵律信息进行有效整合以及对语义与韵律信息进行无缝整合。

最终，本文通过实验验证了 rhyme2vec和层次注意力机制变分自编码器框架训练得到的表征向量在检索和分类等任务上的有效性，实验包括下一行预测、流派分类、歌词生成。通过与现有的一些表现较好的韵律文本学习方法进行比较，最终结果显示 rhyme2vec和层次注意力机制变分自编码器框架相对于这些方法更为有效。

关键词： 韵律文本学习；注意力机制；变分自编码器；表示学习

Abstract

Rhyme text is a special type of text. Compared with common natural language, prosodic text analysis and learning are worthy of specified study. Common rhyme texts mainly include poetry, lyrics, and lyrics in some other art forms. Unlike common natural language, rhyme texts usually have certain prosodic features. The rhymes of rhyme texts are their main feature. In addition to their semantic features, rhyme features need to be paid attention to.

Most of existing rhyme learning methods require manually provided rhyme features to learn, or even completely ignore the prosodic features of rhyme texts, only considering their semantic features and forms.

In order to solve the above problems, this paper proposes a new method, namely rhyme2vec, to learn the prosodic representation vector. This method consists of two models, continuous line rhyme and interlaced rhyme. By integrating these two models, rhyme2vec can handle many features of prosodic patterns well.

This paper also proposes a framework of variational autoencoders combined with hierarchical attention mechanisms to fuse prosodic features and semantic features of rhyme texts. The framework is designed to deal with the problem of representation learning of rhyme texts, incorporating a variety of unexplored mechanisms, namely, the use of attentional mechanisms for the effective integration of prosodic information and the seamless integration of semantic and prosodic information.

In the end, the experiment verifies the effectiveness of the representation vectors obtained by rhyme2vec and the hierarchical attentional mechanism based variational autoencoder framework on tasks such as retrieval and classification. The experiments include the next-line prediction, genre classification, and lyrics generation. By comparing with some state-of-the-art prosodic text learning methods, the final results show that rhyme2vec and the hierarchical attentional mechanism based variational autoencoder frameworks outperform these methods.

Key Words: Rhyme Texts Learning; Attention Mechanism; Variational Autoen-

coder; Representation Learning

目录

第一章 绪论

第一节 研究背景

韵律文本是一种特殊类型的文本，相对于通俗的自然语言，韵律文本分析与学习值得进行专门的研究。常见的韵律文本主要包括诗歌（本文唐诗、宋词、元曲及国外诗歌等统称诗歌）、歌词和一些其他艺术形式中的唱词（如对联以及中国传统艺术中的戏曲、快板等）。与一般的自然语言不同，韵律文本通常是有一定音韵特征的，其中某些韵律文本还有形式上的硬性规定（如诗、词），而在语义上，韵律文本与一般自然语言也有不同，通常韵律文本会利用较为简洁、且与日常语言不同的形式，其形式更为艺术化，用词更为含蓄，相对日常自然语言更为“陌生化”[1]。

韵律文本的韵律是其主要特征。韵律是指相同（或相近）的音素在两个或以上的单词中重复出现。韵律可以出现在同一行中，也可以出现在不同行中。不同类型的韵律文本有着不同的韵律模式要求，如日本的俳句，对于整首作品中的音节数量有十分严格的要求，一首俳句只有三行、十七个音节，形式为五-七-五 [2]。现有的韵律文本学习方法通常需要对于算法适用的韵律文本类型进行较为严格的人为规定，通常某一算法或模型只能适用于一种或少量几种韵律文本。

诗歌是一类很典型的韵律文本，各国的诗歌都有自己的特点。中国诗歌中，古诗词相比于现代诗歌在结构上和韵律上的特征更加明显，如七言诗中每句的平仄基本上只有四种情况（平平仄仄仄平平、平平仄仄平平仄、仄仄平平仄仄平、仄仄平平平仄仄） [3]；英美诗歌中多有音步的规定，如三音步、四音步、五音步等 [4]；日本俳谐连歌中五-七-五-七-七的音节分布规定等 [2]。对于诗歌的分析和学习主要集中在诗歌的生成 [5–10]和诗歌分类 [11, 12]等任务。

歌词是另一种主要的韵律文本形式。与诗歌不同，歌词通常是没有严格的形式限制的，但一般歌词要与歌曲的伴奏配合，所以其结构是与伴奏音频相关的。而说唱歌词与其他类型的歌词略有不同，说唱歌词可以脱离伴奏音乐单独存在，所以几乎不受伴奏音频的影响。由于本文关注的是韵律文本，并没有考

虑音频问题，所以本文关注的歌词主要是说唱歌词。说唱音乐是众多音乐流派中最流行的音乐流派之一 [13]，值得以统计的方法进行探索。学习说唱歌词是一个值得研究的课题，正在受到越来越多学者的关注。而且，说唱歌词学习也是很多实际应用的基础，例如歌词生成 [14, 15]、音乐信息检索 [15]、韵律模式识别 [16, 17]等。

近些年，围绕着说唱歌词分析已经进行了很多研究。这些研究可以被分类为：文本分析 [14]、韵律模式检测 [17, 18]、说唱歌词生成 [14, 19]和评估方法研究 [15, 18, 20]。但是，对于用户来说，这些研究并不足够有效，因为他们要么只是用到了部分的特征，如语义特征；要么并没有学习到足够有效的特征表征，如统计表征。这些研究并不能得到泛用的、同时包含语义和韵律信息的表征。

综上，韵律文本的分析与学习的问题主要有：

- 同一般的自然语言处理（natural language processing，以下简称 NLP）问题一样，韵律文本分析与学习的数据是非结构化的，相对于结构化数据，非结构化数据中难以获取计算机需要的形式语义，因此通过计算机进行处理和分析有很大难度；
- 尽管韵律文本属于自然语言，但现有的 NLP技术由于没有考虑韵律和结构的特征，并不适合直接应用于韵律的分析 [19]；
- 不同类型的韵律文本的结构和韵律特征差异较大，一种算法难以广泛应用于大部分的韵律文本学习任务；
- 从韵律文本中获取文本的韵律特征。计算机直接从文本中无法得到文本的读音，因此无法直接得到文本的韵律特征；
- 韵律特征和语义特征的结合。若已经从韵律文本中获取到了文本的韵律特征和语义特征，那么如何将这两类特征进行融合将是一个挑战，融合的方式、融合时两类特征所占比重都需要进行调试；
- 许多韵律文本学习方法并没有充分利用韵律文本的特征，许多都只是单纯利用了韵律文本的语义特征或韵律特征，没有把二者结合分析；
- 许多韵律文本学习方法是针对特定类型的韵律文本进行的，泛用性很差。

本文利用了变分自编码器（variational autoencoder，以下简称 VAE）来进行韵律文本学习。VAE是一种生成模型，但也可用于数据的特征向量学习与降维。通过实验，VAE [21]已经被证实具有很强的密度建模和生成学习的能力 [22]。在与韵律文本较为相关的音乐信息检索（music information retrieval，以下简称

MIR）领域（与歌词学习相关），一些研究已经将 VAE用作一种实用的工具 [23, 24]。然而，这些研究都只关心很短的音乐旋律片段。本质上，以上工作中的VAE都是用来将音乐特征而非文本特征进行降维。

第二节 本文工作与贡献

本文面向韵律文本，基于现有的方法，进行了改进和结合。本文研究的问题主要包括：

- **韵律文本的韵律特征学习**。现有的韵律文本韵律学习方法主要是基于韵律的统计特征，且需要大量的人工操作和专业知识。 [25]通过统计文本中每行的韵律数量、全文的音节数等统计特征作为文本的韵律特征， [15]同样利用人为规定的一些统计学特征（如文本尾韵的数量、隔行尾韵的数量、行内韵的数量等）进行学习。虽然这些统计学方法可以得到一些可以接受的结果，但效果并不好，且由于其特征需要人为规定，最终学习到的特征有可能并不全面，且限制了模型在不同类型韵律文本之间的迁移能力。另外，韵律文本有多种韵律模式，分别学习这些韵律模式再将其融合到一起可以得到更好的效果；
- **韵律文本整体特征的学习**。学习韵律文本整体特征有两大类方法——同时学习韵律特征和语义特征，直接得到韵律文本的整体特征；分别学习韵律特征和语义特征单独得到韵律文本的韵律特征和语义特征之后，需要将二者融合来得到韵律文本整体的特征。考虑到后者可以将模型模块化，本文选择了后者的学习方法。不同特征融合的方法很多，最简单的可以将两种特征向量直接相加或者拼接，但这两种方法没有考虑到不同特征的权重，尽管这两种方法有效，但还有提升的空间。在 第五章中，本文会展示引入权重对特征融合的提升；
- **韵律特征学习方法效果验证**。本文在两个不同的任务（检索和分类）上，通过与其他韵律特征学习方法进行对比，对本文提出的韵律特征学习方法进行效果验证，对比方法主要是统计学韵律特征。另外本文还需要验证学习同一文本的不同韵律模式并进行综合，要比仅学习某一种韵律模式效果更好，即不同韵律模式之间互为补充；
- **韵律文本整体特征学习方法效果验证**。本文在三个不同的任务上，通过与其他韵律特征学习方法进行对比，对本文提出的韵律特征学习方法进行效

果验证，对比方法包括统计学特征和深度学习得到的表征向量。通过在不同任务上进行对比实验来验证本文模型的泛化能力。第五章对所有对比实验进行了介绍。

与之前的研究不同，本文的研究同时引入了更多特征，即语义特征和韵律特征。相比于简单的语义角度，本文的目的在于为说唱歌词生成一个增加了韵律信息的表示向量。语义信息是通过最先进的段落嵌入方法，doc2vec [26]，将输入的文本数据编码成向量；而对于韵律信息，本文提出了一个新的方法——rhyme2vec，通过整合多种韵律模式来将韵律信息编码成向量。本文还引入了一个基于 VAE的特征融合方法来正确地融合语义信息和韵律信息，以得到一个韵律加强的表征向量。另外，本文还用了注意力机制来平衡多种信息之间的重要性。所有的这些策略被整合成为了一个可泛用的表征学习框架，即层次注意力变分自编码网络（hierarchical attention variational autoencoder network，以下简称HAVAE）。通过实验验证，HAVAE的性能远优于现有的最先进的方法。

本文的贡献在于：

- 本文提出了一个新方法，即 rhyme2vec，来学习韵律表征向量。这个方法包含两个模型，即连续行韵律和隔行韵律。通过整合这两个模型，rhyme2vec可以很好地处理韵律模式的多种特征。
- 本文提出了一个基于 VAE的框架，即 HAVAE。该框架旨在处理韵律文本的表征学习问题。HAVAE整合了多个未被探索的机制，即，韵律信息的有效整合以及对语义与韵律信息的无缝整合。
- 本文通过三个任务（下一行预测、歌曲生成和流派分类）在标准数据集上评估了本文提出的模型，即 HAVAE和 rhyme2vec。实验结果显示，本文提出的模型远优于现有的方法。

第三节 组织结构

本文余下章节的组织结构如下： 第二章主要介绍了目前与韵律文本分析与学习相关的任务及其进展，并且介绍了当前 VAE模型的应用情况； 第三章主要介绍了与本文模型相关的一些预备知识，包括本文主要用到的符号、韵律相关知识、VAE相关知识以及 doc2vec相关知识； 第四章介绍本文提出的模型HAVAE，首先介绍了特征提取模型，包括韵律特征提取模型，即 rhyme2vec，和语义特征提取模型，最后介绍了特征融合模型； 第五章通过实验验证了

rhyme2vec和 HAVAE模型训练得到的表征向量在检索和分类等任务上的有效性，这一章的实验包括下一行预测、流派分类、歌词生成； 第六章总结全文，并对后续工作进行展望。

第二章 研究现状

韵律文本是一类特殊文本，文本除去语义特征之外，还有明确的韵律特征。当前主要的韵律文本分析与学习相关的工作包括诗歌分析与学习 [5–10]和歌词分析与学习 [16, 19, 27, 28]。

第一节 诗歌分析与学习

韵律文本分析中一个典型的任务是诗歌分析。诗歌是一种高度多样化的结构化文学形式。每种诗歌都要遵循它特有的结构模式、韵律模式和音调模式。目前关于诗歌的机器学习方法主要集中在诗歌生成任务。对于诗歌的生成，主要包括基于模板、基于模式、基于遗传算法和基于统计机器翻译等传统的机器学习方法，以及较新的基于深度学习技术的方法。

2.1.1 基于传统机器学习方法的诗歌生成

2.1.1.1 基于模板的方法

基于模板的方法主要用于诗歌生成，模型将现有的诗歌作为模板，将其中一些词句剔除，然后通过将其他新的词语填入这些剔除后的空缺处来生成新的诗歌；或是给出一个规定了诗歌的节数、每节的行数以及每行的音节（单词数量）。Oliveira等人搭建了一个典型的基于模板的诗歌生成模型，其利用一个语义图和一个语法处理器来生成句子，再利用某种生成策略、基于特定的诗歌模板来生成完整的诗歌 [5]。基于模板的方法灵活性很差，且非常依赖人工，一种模板生成的诗歌形式固定，如果需要生成不同类型的诗歌，则需要不同模板。

2.1.1.2 基于模式的方法

基于模式的方法虽然同样是通过事先确定的模式生成诗歌，但相较于基于模板的方法，提高了灵活性。Kurzweil等人的 Cybernetic Poet系统是一个基于模式的方法，该方法以现有诗歌为基础，学习诗歌的词汇、词汇结构、韵律模式以及行文结构等特征，对已有诗歌建模后，进行新诗歌的生成 [6]。

2.1.1.3 基于遗传算法的方法

基于遗传算法的诗歌生成模型由生成模块和评价模块两部分组成。生成模块根据词法、句法、概念等信息产生备选诗作,评价模块则依据一定的准则对备选诗作给予等级评价。蒋锐滢等人的模型采用了遗传算法，其生成模块通过宋词分析句法规范性、语义关联度、不同词牌平仄韵律等特征来生成宋词，评价模块根据句法合法性、主题相关性等标准选取合适的宋词等进行下一步学习，在迭代一定次数后，选取出最终的生成结果 [7]。

2.1.1.4 基于机器翻译的方法

基于机器翻译的诗歌生成方法是利用机器翻译的方法来进行诗歌生成。He Jing等人利用一个统计机器翻译模型，将诗歌的前一句看作源语言、后一句看作目标语言，并添加了平仄押韵等约束来生成后一句，重复进行这一过程，最终得到一首完整的诗歌 [8]。

2.1.1.5 基于传统机器学习方法的问题

从以上方法可见，由于人工的参与，传统机器学习对于诗歌模式的服从程度较高，但也因此十分依赖人工参与，需要设计者有一定的诗歌相关的专业知识，且迁移能力较差，一个模型只能生成一类诗歌（如 [7]的模型只能用于某种词牌的宋词生成）。

2.1.2 基于深度学习方法的诗歌生成

基于深度学习的方法相对于基于传统机器学习的方法，迁移能力更强，同一个模型可以适应更多种类的诗歌，甚至可以不受诗歌种类的限制。如Mikolov等人采用了一个循环神经网络（recurrent neural network，以下简称RNN），给定初始的诗句，然后根据输入的诗句生成新的诗句，当前的输出又会是下一次的输入，多次重复后即可生成完整的诗歌 [9]；Zhang Xingxing等人利用一个句子级别的 CNN和两个分别用于字符和句子级别的 RNN构建了一个中文诗歌的生成模型 [10]。基于深度学习的诗歌生成方法在一定程度上克服了传统机器学习在迁移能力方面的局限，但没有考虑到韵律特征，并不能对诗歌进行完全的学习。

第二节 歌词学习研究现状

歌词也是韵律文本的一类，与诗歌不同，歌词基本上没有固定的形式，某些诗歌学习方法便没有办法直接应用于歌词学习中，例如基于模板的诗歌生成方法。近来，研究者们开始研究歌词主要的工作包括歌词量化评分、歌词分类、歌词生成等任务。

2.2.1 歌词量化评分

评价一首歌词的好坏是一个较为主观的问题，不同的人可能会有不同的喜好。然而，由于歌词生成等任务的需要，一些研究人员在歌词的量化评分方面做了许多研究。Hirjee等人作者们提出了一个基于说唱歌词中音素频率的概率评分模型。他们的模型可以自动检测内含的和行末的韵律，但是它需要额外的人工标记的说唱歌词和韵律对 [16]。Malmi等人提出了一种名为韵律密度（rhyme density）的歌词量化评分方法，该方法通过计算所有单词和相近单词中匹配元音音素个数的平均数来评价一首歌词的韵律质量。

2.2.2 歌词分类

早期的歌词分类方法多基于统计方法，或通过人为规定一些特征来进行分类。Hu Xiao等人主要用到了词袋模型（bag-of-words）的方法构建歌词的文本特征向量进行歌词的分类 [27]。基于 [27]，He Hui等人提出了一些人工定义的文本特征，利用这些文本特征学习文本特征向量进行歌词分类 [28]。这些方法主要基于统计特征或者人工规定的特征，并不是非常有效， 第五章第四节中将展示一种基于统计特征的歌词学习方法。

2.2.3 歌词生成

相比于诗歌生成，歌词生成在长短、韵律等形式上所受到的限制更少，因此，所用的方法也有所不同，许多用于诗歌生成的方法都不适用于歌词生成任务（如基于模板的方法、基于模式的方法等）。

Dekai Wu等人提出了一个用于说唱对决即兴创作的生成任务。他们将即兴创作看作是类似翻译的任务，这样，任何挑战歌词都会被翻译为回应的歌词；然而，模型训练之前，需要一个“翻译词典” [19]。Potash等人应用了长短时记忆网络（long short-term memory network，以下简称 LSTM），模仿特定说唱歌手生成歌词；然而，为了捕捉韵律模式，这个模型需要足够的训练

数据，而且韵律对必须在语料库中足够频繁地出现 [14]。目前最突出的研究是 Malmi等人的模型，称为 DopeLearning。作者引入了 EndRhyme、EndRhyme-1和 OtherRhyme来作为韵律特征，引入 LineLength作为结构特征，并且从 BOW、BOW5、LSA和 NN5中提取语义特征 [15]。

在现有的研究中，DopeLearning最有效地提取了说唱歌词特征。尽管NN5将字符级别的向量作为输入，但说唱歌词的最终表征仅仅加入了模型输出的置信度。因此，DopeLearning学习到的表征向量只是一些统计学特征的组合。另外，DopeLearning的韵律特征中，只考虑了元音音素；而且所有特征都是线性计算得到的，学习能力有限。

第三节 VAE的研究现状

VAE作为一种深度学习网络，在许多应用领域（如计算机视觉、NLP等领域）都较为有效。VAE网络既可以用作表征学习网络，也可以用作生成模型。

2.3.1 VAE作为生成模型的应用

计算机视觉领域主要将 VAE当做一种生成模型，从一个特定维度的向量生成某一类图片。Cai Lei等人设计了一个多阶段的 VAE模型，该模型以图片为输入，先利用 VAE粗糙地重构图片，再在这张粗糙的图片的基础上重构出清晰的图片 [29]。

在 NLP领域，VAE同样可以作为一种生成模型。与图片不同，单词属于离散型的数据，生成文本时，模型的损失函数优化会有困难。Semeniuta等人设计了一个基于 CNN和 RNN的 VAE模型，用于从连续空间中生成句子，为了便于优化，该模型用一个辅助重构项替换了 VAE损失函数中的 $\mathcal{KL}$ 散度项 [30]。

在 MIR领域，一些基于 VAE的框架被用来解决特定的一些问题。在 [31]中，作者提出了一个变分循环自编码器（variational recurrent autoencoder，以下简称 VRAE）来生成视频游戏的背景音乐。Alexey等人的工作中提出了一种历史信息支持的 VRAE（variational recurrent autoencoder supported by history，以下简称 VRASH）来生成单音调音乐 [23]。在 [24]中，作者为 VAE提出了一种精炼的正则化函数，用于生成复调合唱旋律。以上的这些方法都只专注于音乐旋律，而非歌词，而且 VAE也只是用来学习单纯的音频特征。

2.3.2 VAE用于表征学习

本文中利用 VAE作为一个表征学习模型，将韵律文本输入一个基于 VAE的模型，学习到一个最终的表征向量。VAE学习到的表征向量可以用于如分类等多种任务。Xu WeiDi等人设计了一个基于 RNN的 VAE模型，模型采用了半监督的方式，首先用无标签数据在 VAE中学习数据的表征向量，之后再用模型计算出有标签数据的表征向量进行分类 [32]。

第四节 本章小结

本章介绍了诗歌分析与学习、歌词分析与学习和基于 VAE模型的研究现状。诗歌与歌词是两类典型的韵律文本。现有的诗歌和歌词学习方法中，要么只利用了传统的机器学习方法，迁移能力差，要么没有有效地利用到所有的特征（如只利用到语义特征，而没有利用韵律特征）。关于 VAE模型，本章介绍了 VAE用于生成模型和表征学习的研究现状，并着重介绍了 VAE在 MIR领域和 NLP领域的发展和应用情况。在 NLP领域，VAE模型只是用于纯文本的分析与学习，仍没有运用于韵律文本的 VAE模型；在 MIR领域，VAE主要用于音乐生成，并没有应用于歌词学习。

第三章 模型相关知识

在介绍 HAVAE模型之前，本章将先介绍一下模型相关的预备知识，包括所用的符号、韵律相关知识、VAE相关知识和 doc2vec相关知识等。

本节将首先介绍一些本文中即将出现的符号。$\boldsymbol{L}$ 表示一个由 n 行说唱歌词组成的序列。为了便于描述，本文用表 3.2中的一个说唱歌词片段的例子来直观地展示模型的主要想法、解释本文提出的技术。表中的例子是从 Fort Minor的《Remember the Name》中节选的4行连续的歌词。令 $\boldsymbol{L_s}$ 表示 $\boldsymbol{L}$ 的原始文本，如表 3.2中左半部分所示；$\boldsymbol{L_r}$ 表示 $\boldsymbol{L}$ 相应的韵律形式，如表 3.2中右半部分所示。本文中用 $\boldsymbol{v_s}$ 表示从 $\boldsymbol{L_s}$ 编码得到的语义向量，用 $\boldsymbol{v_r}$ 表示从 $\boldsymbol{L_r}$ 编码得到的韵律向量。$\boldsymbol{v_s}$ 和 $\boldsymbol{v_r}$ 的维度分别为 d_s 和 d_r。$\boldsymbol{v_t}$ 表示从 $\boldsymbol{L}$ 学习到的目标特征向量。N 表示输入样例的数量。

第一节 韵律文本相关知识

英语有48个国际音标[1]。我们将破擦音（如，[ʧ]）和双元音（如，[ɔɪ]）看作两种不同的音素。本文用 eSpeak[2]将说唱歌词翻译成了其对应的音素符号。eSpeak是一款可用于多种语言的紧凑型开源语音合成器。 eSpeak有命令行和带GUI的版本，可用于从文件或标准输入流中接收输入文本，并生成对应的声音。为了从文本生成声音，eSpeak也附带生成了文本对应的音素符号。最终的音素符号文件是特定音素字母表中的一系列字符组成的序列，字母表中，每一个音素被表示为一个字母。本文主要考虑英文，英文音素的字母表如下所示：

韵律是指相同（或相近）的音素在两个或以上的单词中重复出现[3]。韵律可以出现在同一行中，也可以出现在不同行中。在说唱歌曲中，韵律占有非常重要的地位。韵律有两种主流的模式，即单韵和隔行韵。单韵指在连续几行中，每行都有相同的尾韵。隔行韵指在连续几行中，奇数行和偶数行各自分别有相同的尾韵。单韵、隔行韵以及其他韵律模式在整首的说唱歌词里会随机出现，

[1] https://en.wikipedia.org/wiki/International_Phonetic_Alphabet
[2] http://espeak.sourceforge.net/
[3] https://en.wikipedia.org/wiki/Rhyme

表 3.1 主要符号定义

符号	含义
$\boldsymbol{L}$	一个由若干行韵律文本组成的序列
$l_i, i=1,2,...,n$	$\boldsymbol{L}$ 的第 i 行韵律文本
$\boldsymbol{L_s}$	$\boldsymbol{L}$ 的原始文本
$l_{s,i}, i=1,2,...,n$	$\boldsymbol{L_s}$ 的第 i 行
$\boldsymbol{L_r}$	$\boldsymbol{L}$ 相应的音素符号序列
$l_{r,i}, i=1,2,...,n$	$\boldsymbol{L_r}$ 的第 i 行
$\boldsymbol{v_s}$	从 $\boldsymbol{L_s}$ 学习到的语义向量
$\boldsymbol{v_r}$	从 $\boldsymbol{L_r}$ 学习到的韵律向量
$\boldsymbol{v_t}$	模型最终学习到的 $\boldsymbol{L}$ 的表征向量
$\boldsymbol{L_r^m}$	$\boldsymbol{L_r}$ 的所有连续行（$\boldsymbol{L_r^m}=\boldsymbol{L_r}$）
$\boldsymbol{L_r^o}$	$\boldsymbol{L_r}$ 中的奇数行
$\boldsymbol{L_r^e}$	$\boldsymbol{L_r}$ 中的偶数行
$\boldsymbol{v_r^m}$	从 $\boldsymbol{L_r^m}$ 学习到的单韵韵律特征向量
$\boldsymbol{v_r^o}$	从 $\boldsymbol{L_r^o}$ 学习到的隔行韵律特征向量
$\boldsymbol{v_r^e}$	从 $\boldsymbol{L_r^e}$ 学习到的隔行韵律特征向量

并且有不同的重要性。考虑到单韵和隔行韵在说唱歌词中出现最为频繁，本文只考虑单韵和隔行韵

第二节 问题提出

3.2.1 韵律特征提取问题

本文假设所有说唱歌词都同时包含单韵和隔行韵。对于单韵，我们把所有连续行看作一个韵律段 $\boldsymbol{L_r^m}$。对于隔行韵，本文将一首说唱歌词分成两个韵律段，一个只包含奇数行 $\boldsymbol{L_r^o}$（表 3.2中的红色行）；另一个只包含偶数行 $\boldsymbol{L_r^e}$（表 3.2中蓝色行）。因此，每个样例都对应3个韵律段，对于所有输入样例，总共有 $3N$ 个韵律段。韵律表征学习问题的定义如下：

表 3.2 说唱歌词样例及其对应的韵律形式

原始文本 (L_s)	韵律形式 (L_r)
Put it together himself	p,Ut It t@g,ED3 hIms'Elf
now the picture connects	n'aU D@ p'IktS3 k@n'Ekts
Never asking for someone's help	n'Ev3r- 'aaskIN fO@ s'Vmw0nz h'Elp
to get some respect	t@ gEt s,Vm rI2sp'Ekt

表 3.3 音素字母表

元音	[@][3][3:][@L][@2][@5][a][aa][a#][A:][A@][E][e@][I][I2][i][i:][i@] [0][V][u:][U][U@][O:][O@][o@][aI][eI][OI][aU][oU][aI@][aU@] [u:][U][U@][O:][O@][o@][aI][eI][OI][aU][oU][aI@][aU@]
辅音	[p][b][t][d][tS][dZ][k][g][f][v][T][D][s][z][S][Z][h][m][n][N][l][r][j][w]

问题 1 给定一首 n 行的说唱歌词 $\boldsymbol{L}$，$\boldsymbol{L_r}$ 为对应的韵律形式。令 $\boldsymbol{v}_r^m$ 表示 $\boldsymbol{L}_r^m$ 的韵律向量，$\boldsymbol{v}_r^o$ 表示 $\boldsymbol{L}_r^o$ 的韵律向量，$\boldsymbol{v}_r^e$ 表示 $\boldsymbol{L}_r^e$ 的韵律向量。目标是通过融合 $\boldsymbol{v}_r^m$、$\boldsymbol{v}_r^o$ 和 $\boldsymbol{v}_r^e$，学习到 $\boldsymbol{v_r}$。

为了解决问题 1，本文在第四章第一节提出了一个新的方法——rhyme2vec。rhyme2vec模型包含两个模块，一是连续行模块，称为 C-Line，用来处理单韵部分，最终得到 $\boldsymbol{v}_r^m$；另一个是跳行模块，称作 Skip-Line，用来处理隔行韵部分，最终得到 $\boldsymbol{v}_r^o$ 和 $\boldsymbol{v}_r^e$。

3.2.2 特征融合问题

一首好的说唱歌曲一定有优秀的主题和有吸引力的韵律。因此，说唱歌词的表征向量应该同时包含语义信息和韵律信息。说唱歌词表征学习定义如下：

问题 2 给定一首 n 行的说唱歌词 $\boldsymbol{L}$，假设韵律向量 $\boldsymbol{v_r}$ 和语义向量 $\boldsymbol{v_s}$ 已知，通过融合 $\boldsymbol{v_r}$ 和 $\boldsymbol{v_s}$ 中有用的信息学习到目标向量 $\boldsymbol{v}_t$。

本文在第二节提出了基于 VAE的特征融合模块，用以融合 $\boldsymbol{v_s}$ 和 $\boldsymbol{v_r}$。融合模块中还引入了注意力机制来学习 $\boldsymbol{v_s}$ 和 $\boldsymbol{v_r}$ 之间的相互关系。目标表征向量 $\boldsymbol{v_t}$ 是通过在一个隐含高斯分布中采样得到的。本文提出的模型的输出，即 $\boldsymbol{v_t}$，可以被用来处理许多与说唱歌词有关的任务，其中一些将在第五章展示。

第三节 变分自编码器（VAE）

VAE是一种生成模型，但在模型的训练过程中，可以对数据的进行编码，所以也可以用来学习表征向量。

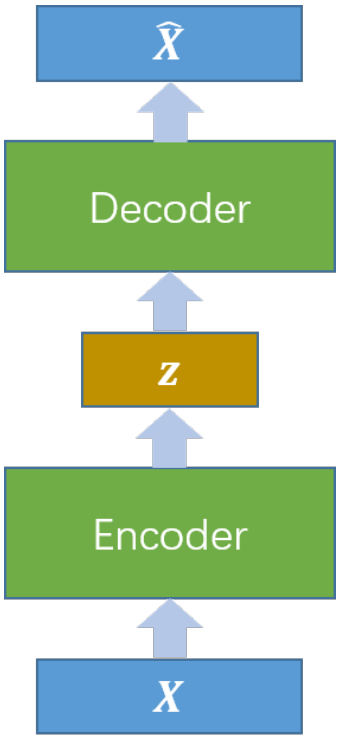

图 3.1 VAE的简化图

VAE的主要结构如图 3.1所示。一个训练好的 VAE可以在给定一个特定维度的输入向量之后，生成一个特定的结果（如一幅图像）。在训练的过程中，需要将现有的数据输入 VAE中给定数据 $\boldsymbol{X}$（$\boldsymbol{X}$ 可以是用向量表示的任意数据）编码成一个低维的表征向量 $\boldsymbol{z}$，然后再从 $\boldsymbol{z}$ 出发，重构出 $\hat{\boldsymbol{X}}$，使得 $\boldsymbol{X}$ 和 $\hat{\boldsymbol{X}}$ 越接近越好。VAE最根本的目标在于最大化从 $\boldsymbol{z}$ 重构 $\boldsymbol{X}$ 的概率，目标如下所示：

$$P(\boldsymbol{X}) = \int P(\boldsymbol{X}|\boldsymbol{z},\theta)P(\boldsymbol{z})d\boldsymbol{z}. \tag{3.1}$$

在生成的过程中，随机选取一个向量 $\boldsymbol{r}$，从图 3.1所示的模型中的 $\boldsymbol{z}$ 处输入，即可从 $\hat{\boldsymbol{X}}$ 得到一个生成的数据。

实际上，对于特定数据 $\boldsymbol{X}$，只会在特征空间中对应特定的 $\boldsymbol{z}$，换句话说，对于绝大多数 $\boldsymbol{z}$，都是无法生成对应的 $\boldsymbol{X}$ 的，即对于绝大多数 $\boldsymbol{z}$，$P(\boldsymbol{X}|\boldsymbol{z},\theta)$ 会接近

于0。在训练的过程中，模型需要找到这个特定的 $\boldsymbol{z}$，使得 $P(\boldsymbol{X}|\boldsymbol{z},\theta)$ 尽量地大，并从中计算 $P(\boldsymbol{X})$。这意味着我们需要一个新的函数 $Q(\boldsymbol{z}|\boldsymbol{X})$，它可以通过输入 $\boldsymbol{X}$ 的值，给出一个可能产生 $\boldsymbol{X}$ 的 $\boldsymbol{z}$ 值的分布。模型会使得 Q值下可能的 $\boldsymbol{z}$ 的空间会大大地小于先前 $P(\boldsymbol{z})$ 下可能的所有 $\boldsymbol{z}$ 的空间。通过这个方法，可以改为计算 $E_{\boldsymbol{z}\sim Q}P(\boldsymbol{X}|\boldsymbol{z})$，计算会变得更为容易。然而，如果 $\boldsymbol{z}$ 不服从 $\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$，而是从一个随机的概率分布 $Q(\boldsymbol{z})$ 中采样的，那么这并不能帮助优化 $P(\boldsymbol{X})$。VAE需要首先把 $E_{\boldsymbol{z}\sim Q}P(\boldsymbol{X}|\boldsymbol{z})$ 和 $P(\boldsymbol{X})$ 联系起来。

$E_{\boldsymbol{z}\sim Q}P(\boldsymbol{X}|\boldsymbol{z})$ 和 $P(\boldsymbol{X})$ 之间的关系是变分贝叶斯方法的基础之一。从 $P(\boldsymbol{X}|\boldsymbol{z})$ 和 $Q(\boldsymbol{z})$ 之间的 Kullback-Leibler散度（$\mathcal{KL}$ 散度，即 $\mathcal{D}$）的定义开始，对于一些任意 Q：

$$\mathcal{D}[Q(\boldsymbol{z})||P(\boldsymbol{z}|\boldsymbol{X})] = E_{\boldsymbol{z}\sim Q}[\log Q(\boldsymbol{z}) - \log P(\boldsymbol{z}|\boldsymbol{X})] \tag{3.2}$$

通过将应用贝叶斯准则应用于 $P(\boldsymbol{z}|\boldsymbol{X})$，可以将 $P(\boldsymbol{X})$ 和 $P(\boldsymbol{X}|\boldsymbol{z})$ 引入公式 3.2：

$$\mathcal{D}[Q(\boldsymbol{z})||P(\boldsymbol{z}|\boldsymbol{X})] = E_{\boldsymbol{z}\sim Q}[\log Q(\boldsymbol{z}) - \log P(\boldsymbol{X}|\boldsymbol{z}) - \log P(\boldsymbol{z})] + \log P(\boldsymbol{X}) \tag{3.3}$$

在这里，$\log P(\boldsymbol{X})$ 被从期望中提出，是因为它不依赖于 $\boldsymbol{z}$。将等式中 $-\log P(\boldsymbol{X}|\boldsymbol{z}) - \log P(\boldsymbol{z})$ 改写成 $\mathcal{KL}$ 散度的形式后得到：

$$\log P(\boldsymbol{X}) - \mathcal{D}[Q(\boldsymbol{z})||P(\boldsymbol{z}|\boldsymbol{X})] = E_{\boldsymbol{z}\sim Q}[\log P(\boldsymbol{X}|\boldsymbol{z})] - \mathcal{D}[Q(\boldsymbol{z})||P(\boldsymbol{z})] \tag{3.4}$$

这里，需要注意的是，$\boldsymbol{X}$ 是固定的，Q 可以是任何分布，而不仅仅是一个将 $\boldsymbol{X}$ 映射到可以产生 $\boldsymbol{X}$ 的 $\boldsymbol{z}$ 的分布。由于现在需要推导 $P(\boldsymbol{X})$，所以构造一个取决于 $\boldsymbol{X}$ 的 Q 很有必要，而且，这个 Q 要使 $\mathcal{D}[Q(\boldsymbol{z})||P(\boldsymbol{z}|\boldsymbol{X})]$ 变小：

$$\log P(\boldsymbol{X}) - \mathcal{D}[Q(\boldsymbol{z}|\boldsymbol{X})||P(\boldsymbol{z}|\boldsymbol{X})] = E_{\boldsymbol{z}\sim Q}[\log P(\boldsymbol{X}|\boldsymbol{z})] - \mathcal{D}[Q(\boldsymbol{z}|\boldsymbol{X})||P(\boldsymbol{z})] \tag{3.5}$$

这个等式相当于 VAE的核心，所以这里用两句话解释一下它的内容。等式左边有 VAE想要最大化的值：$\log P(\boldsymbol{X})$（加上一个误差项 $\mathcal{D}[Q(\boldsymbol{z}|\boldsymbol{X})||P(\boldsymbol{z}|\boldsymbol{X})]$，它使得 $\boldsymbol{Q}$ 产生可以重构给定 $\boldsymbol{X}$ 的 $\boldsymbol{z}$）。而对于等式右边，VAE可以通过选择正确的 Q 来进行随机梯度上升优化（然而不同的 Q 对于结果的影响并不明显）。值得注意的是，VAE模型在这里突然采用了一种看起来像自动编码器（autoencoder，即 AE）的形式（尤其是公式 3.5的右边），因为 Q 将 $\boldsymbol{X}$ 编码为 $\boldsymbol{z}$，并且 P 对其进行“解码”以重建 $\boldsymbol{X}$。

VAE的目标就是最大化公式 3.5。首先看 $\mathcal{D}[Q(\boldsymbol{z}|\boldsymbol{X})||P(\boldsymbol{z})]$。由于 Q 和 P 的具体分布对于结果并没有很大影响，这里假设 Q 和 P 都是高斯分布，即 $Q=$

$\mathcal{N}(\boldsymbol{\mu_0}|\boldsymbol{\Sigma_0})$，$P=\mathcal{N}(\boldsymbol{\mu_1}|\boldsymbol{\Sigma_1})$。这里为了简化问题，假设 P 的分布是标准高斯分布，即 $\mathcal{N}(\mathbf{0}|\boldsymbol{I})$。那么公式 3.5中 $\mathcal{KL}$ 散度的部分即可改写为 $\mathcal{D}[\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X})|\boldsymbol{\Sigma}(\boldsymbol{X}))||\mathcal{N}(\mathbf{0}|\boldsymbol{I})]$。这部分可以通过如下过程进行简化：

$$
\begin{aligned}
&\mathcal{D}[\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X})|\boldsymbol{\Sigma}(\boldsymbol{X}))||\mathcal{N}(\mathbf{0}|\boldsymbol{I})] \\
=&\int[\frac{1}{2}\log\frac{1}{|\boldsymbol{\Sigma}(\boldsymbol{X})|}-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu}(\boldsymbol{X}))^{\top}\boldsymbol{\Sigma}(\boldsymbol{X})^{-1}(\boldsymbol{z}-\boldsymbol{\mu}(\boldsymbol{X}))+\frac{1}{2}\boldsymbol{z}^{\top}\boldsymbol{z}]\times\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{X})|\boldsymbol{\Sigma}(\boldsymbol{X}))dz \\
=&\frac{1}{2}\log\frac{1}{|\boldsymbol{\Sigma}(\boldsymbol{X})|}-\frac{1}{2}\mathrm{tr}\{E[(\boldsymbol{z}-\boldsymbol{\mu}(\boldsymbol{X})(\boldsymbol{z}-\boldsymbol{\mu}(\boldsymbol{X}))^{\top}]\boldsymbol{\Sigma}(\boldsymbol{X})^{-1}\}+\frac{1}{2}E[\boldsymbol{z}^{\top}\boldsymbol{z}] \\
=&\frac{1}{2}\log\frac{1}{|\boldsymbol{\Sigma}(\boldsymbol{X})|}-\frac{1}{2}d+\frac{1}{2}\boldsymbol{\mu}(\boldsymbol{X})^{\top}\mu(\boldsymbol{X})+\frac{1}{2}\mathrm{tr}\{\boldsymbol{\Sigma}(\boldsymbol{X})\} \\
=&-\frac{1}{2}[\log|\boldsymbol{\Sigma}(\boldsymbol{X})|+d-\boldsymbol{\mu}(\boldsymbol{X})^{\top}\mu(\boldsymbol{X})-\mathrm{tr}\{\boldsymbol{\Sigma}(\boldsymbol{X})\}].
\end{aligned}
\tag{3.6}
$$

之后，对于公式 3.5中 $E_{\boldsymbol{z}\sim Q}[\log P(\boldsymbol{X}|\boldsymbol{z})]$ 的部分，需要用到重参数化的技巧。对这部分进行估计可以通过对 $\boldsymbol{z}$ 进行采样。但是，若要取得好的效果，则需要对 $\boldsymbol{z}$ 进行大量的采样，这将是一个非常大的开销。所以，在实际优化过程中，只是对 $\boldsymbol{z}$ 进行一次采样，将这次采样作为对真实分布的估计。目标函数优化的过程中需要用到梯度下降，但是，这就造成了一个严重的问题，那就是，如果采用对 $\boldsymbol{z}$ 采样的方式来优化目标函数，那么最终的梯度和 $\mathcal{KL}$ 散度部分将会没有任何联系，也就是说，用梯度下降的方法将无法优化整个目标函数。这里就需要用到重参数化的方法。

重参数化即在采样时，先从标准正态分布 $\mathcal{N}(\mathbf{0}|\boldsymbol{I})$ 中采样一个随机向量 $\boldsymbol{\varepsilon}$，然后通过 $\boldsymbol{\varepsilon}\bigodot\boldsymbol{\Sigma}(\boldsymbol{X})+\boldsymbol{\mu}(\boldsymbol{X})$ 获得采样得到的 $\boldsymbol{z}$，其中 $\bigodot$ 表示向量之间的按位乘法。这样，$\boldsymbol{z}$ 就和 $\boldsymbol{\Sigma}(\boldsymbol{X})$ 和 $\boldsymbol{\mu}(\boldsymbol{X})$ 有了联系，梯度的传递就可以达到 $\mathcal{KL}$ 散度部分了。

对于不同的数据，$P(\boldsymbol{X}|\boldsymbol{z})$ 对应不同的分布，若数据是实数数据，那么对应的是高斯分布，若数据是0-1数据，那么对应的是伯努利分布。以下讨论最简单的情况，即采用多层感知机（multi-layer perceptrons，以下简称 MLP）构建 VAE。

对于高斯分布的情况，在图 3.3中 Decoder部分需要从 $\boldsymbol{z}$ 再采样 $\boldsymbol{X}$ 来重构

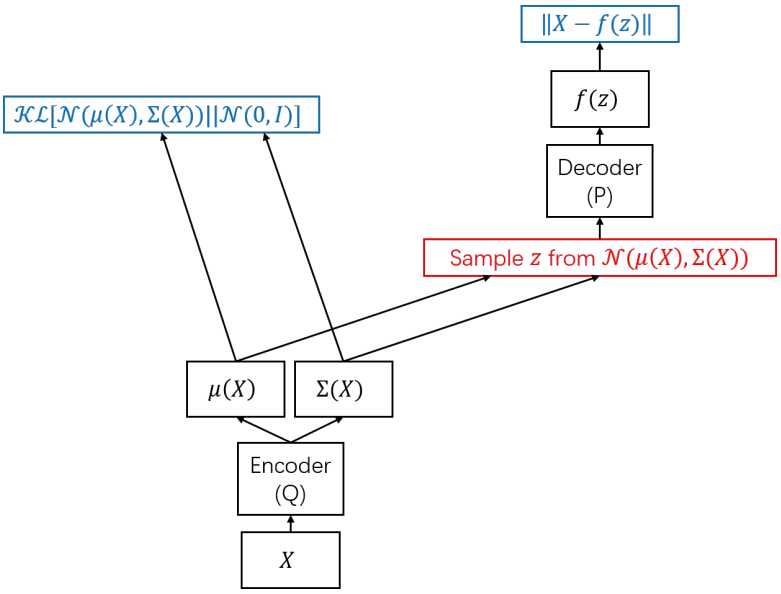

图 3.2 直接采样 z

$\boldsymbol{X}$。$\log P(\boldsymbol{X}|\boldsymbol{z})$ 通过以下方式计算：

$$\begin{aligned} \log P(\boldsymbol{X}|\boldsymbol{z}) &= \log \mathcal{N}(\boldsymbol{X}; \boldsymbol{\mu}(\boldsymbol{z}), \boldsymbol{\Sigma}(\boldsymbol{z})) \\ \boldsymbol{\mu}(\boldsymbol{z}) &= \boldsymbol{W_1 h} + \boldsymbol{b_1} \\ \log \boldsymbol{\Sigma}(\boldsymbol{z}) &= \boldsymbol{W_2 h} + \boldsymbol{b_2} \\ \boldsymbol{h} &= f(\boldsymbol{W_3 z} + \boldsymbol{b_3}), \end{aligned} \tag{3.7}$$

其中，$\boldsymbol{W_1}$、$\boldsymbol{W_2}$、$\boldsymbol{W_3}$、$\boldsymbol{b_1}$、$\boldsymbol{b_2}$ 和 $\boldsymbol{b_3}$ 都是全连接层的参数，f 是全连接层后的激活函数。

对于伯努利分布的情况，图 3.3中 Decoder部分就是 MLP，$\log P(\boldsymbol{X}|\boldsymbol{z})$ 通过以下方式计算：

$$\begin{aligned} \log P(\boldsymbol{X}|\boldsymbol{z}) &= \sum \boldsymbol{X_i} \log \hat{\boldsymbol{X}}_{\boldsymbol{i}} + (1 - \boldsymbol{X_i}) \log(1 - \hat{\boldsymbol{X}}_{\boldsymbol{i}}) \\ \hat{\boldsymbol{X}}_{\boldsymbol{i}} &= sigmoid(\boldsymbol{W_5}(\boldsymbol{W_4 z} + \boldsymbol{b_4}) + \boldsymbol{b_5}), \end{aligned} \tag{3.8}$$

第四节 doc2vec模型

doc2vec模型是一种学习文本表征向量的模型，通过学习文本中不同单词的分布情况得到整段文本的表征向量。

doc2vec模型基于 word2vec模型——一种单词表征学习模型。word2vec模型

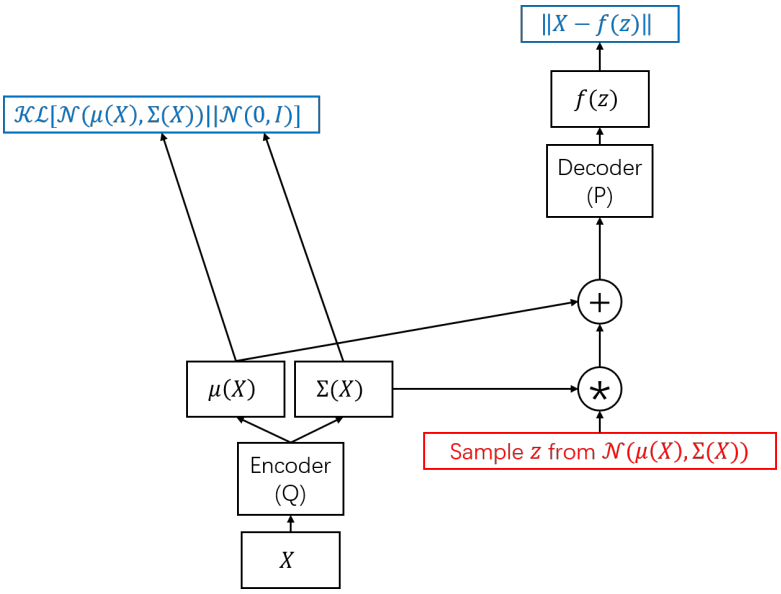

图 3.3 采用重参数化

如图 3.4所示。word2vec利用一个滑动窗口扫过整个文本，每次取窗口中的所有单词的表征向量进行更新，图 3.4中 CBOW模型的目标为最大化公式 3.9，

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(\boldsymbol{w_k}|Context(\boldsymbol{w_k})), \tag{3.9}$$

其中 T 为整段文本的单词总数，k 为滑动窗口一侧的大小，$\boldsymbol{w_i}$ 为第 i 个单词的表征向量，$Context(\boldsymbol{w_k})$ 是单词 k 的上下文中单词的向量。图 3.4中 Skip-gram模型的目标为最大化公式 3.10，

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(Context(\boldsymbol{w_k})|\boldsymbol{w_k}), \tag{3.10}$$

最大化公式 3.9和公式 3.10就是最大化在每个窗口中单词共同出现的概率。

与 word2vec模型类似，doc2vec模型也有两种模式，如图 3.5所示，分别为 PV-DM和 PV-DBOW。

doc2vec模型相比于 word2vec模型，加入了一个文本标签向量。文本标签向量在训练时与其他单词向量一起训练，但是与其他单词有一点不同，即文本标签向量在每个窗口训练时都会更新。PV-DM模型类似于 word2vec的 CBOW模型，PV-DBOW类似于 word2vec的 Skip-gram模型。

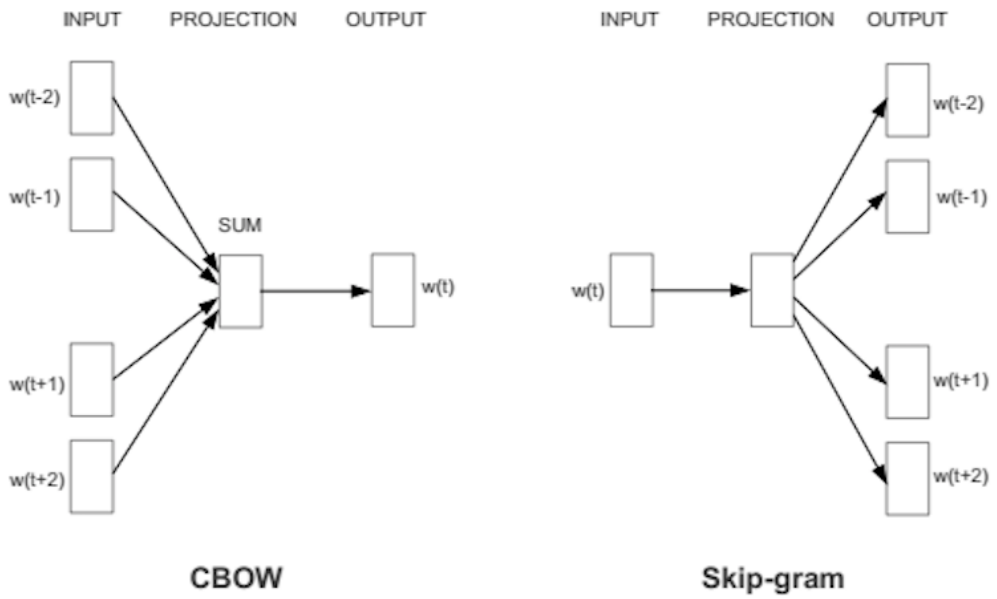

图 3.4 word2vec的两种模式

word2vec模型在学习时有两种优化方式，一是层次 softmax [33]，二是负采样 [34]。

层次 softmax方法基于哈夫曼树。如图 3.6所示，哈夫曼树的每个节点都对应一个向量，叶节点对应的是单词向量，而内部节点对应的是一个辅助向量 $\boldsymbol{\theta}_i^{\boldsymbol{w}}$，映射层中 $\boldsymbol{x}_w$ 为输入层各个向量的和。在层次 softmax中，每一层节点都相当于一次逻辑回归的二分类。从根节点到达一个叶节点的路径，即为路径上每个公式 3.9中 $p(\boldsymbol{w_k}|\boldsymbol{w}_{t-k},...,\boldsymbol{w}_{t+k})$ 即为从根节点开始，到 $\boldsymbol{w_k}$ 对应的叶节点路径上二分类概率之积。以图 3.6中单词 girl为例，从根节点出发，到 girl的节点，经过了 d_2、d_3、d_4、d_5 四个节点，每次二分类的概率分别为：

$$\begin{aligned} p(d_2|\boldsymbol{x_w},\boldsymbol{\theta_1^w}) &= \sigma(\boldsymbol{x_w^\top}\boldsymbol{\theta_1^w}) \\ p(d_3|\boldsymbol{x_w},\boldsymbol{\theta_2^w}) &= 1-\sigma(\boldsymbol{x_w^\top}\boldsymbol{\theta_2^w}) \\ p(d_4|\boldsymbol{x_w},\boldsymbol{\theta_3^w}) &= 1-\sigma(\boldsymbol{x_w^\top}\boldsymbol{\theta_3^w}) \\ p(d_5|\boldsymbol{x_w},\boldsymbol{\theta_4^w}) &= \sigma(\boldsymbol{x_w^\top}\boldsymbol{\theta_4^w}), \end{aligned} \tag{3.11}$$

从而 $p(\boldsymbol{girl}|Context(girl)) = \prod_{i=2}^{5} p(d_i|\boldsymbol{x_w},\boldsymbol{\theta}_{i-1}^{\boldsymbol{w}})$，其中 σ 是 sigmoid函数。Skip-gram的计算方式与 CBOW类似，由于本文采用 doc2vec的 PV-DM模型，与

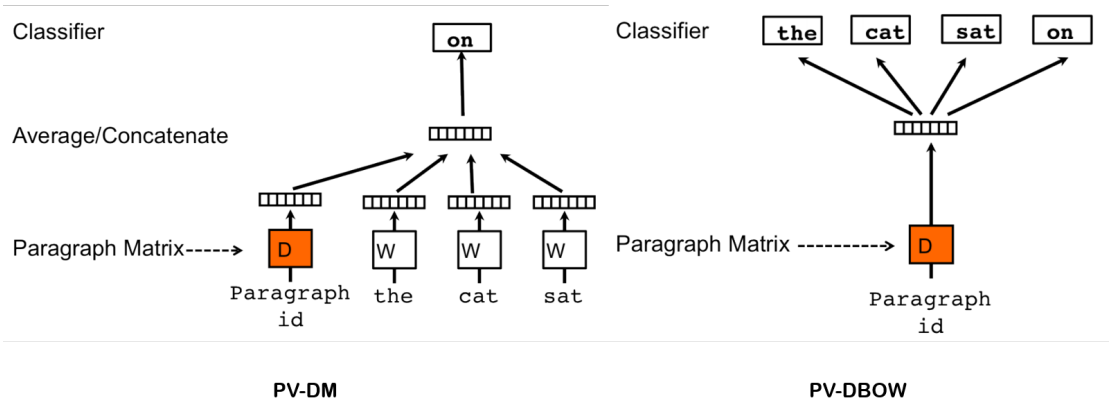

图 3.5 word2vec的两种模式

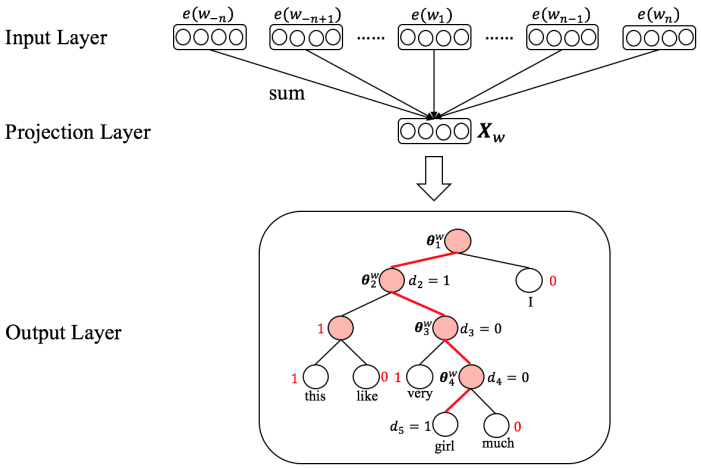

图 3.6 hierarchical softmax

word2vec的 CBOW模型接近，所以此处对于 Skip-gram的计算方式不再赘述。

在负采样的优化方法中，目标函数变为

$$p(\boldsymbol{w_k}|Context(\boldsymbol{w_k})) = \prod_{u \in k \bigcup Neg(k)} [\sigma(\boldsymbol{x_w^\top w_u})]_k^T(u) \cdot [1 - \sigma(\boldsymbol{x_w^\top w_u})]^{[1-T_k(u)]}, \quad (3.12)$$

其中，$Neg(k)$ 是对单词 k 进行负采样得到的负样本集合；u 是对单词 k 采样得到的样本单词的序号；$T_k(u)$ 是一个标志量，若 $u = k$，则 $T_k(u) = 1$，若 $u \neq k$，则 $T_k(u) = 0$。

第五节 本章小结

本章介绍了与 HAVAE模型相关的一些预备知识。

本章首先介绍了本文中主要用到的一些符号；之后介绍了一些韵律文本相关知识，包括单韵和隔行韵等韵律模式、模型用到的音素字母表等；之后提出了韵律特征提取问题和特征融合问题；之后简单介绍了 VAE的原理及其训练过程；最后介绍了 doc2vec模型。

第四章 利用层次注意力机制的 VAE模型

本文提出的模型的结构如图 4.1所示。模型包括两个主要模块，分别是特征提取模块和特征融合模块。特征提取模块用于将输入内容编码成分布式的韵律和语义向量。输入内容包括原始的说唱歌词，以及歌词的韵律形式。特征融合模块用于有效地融合语义信息和韵律特征。最终学习到的表征向量可以在许多任务中代表说唱歌词，例如本文实验部分将提到的歌词预测任务。

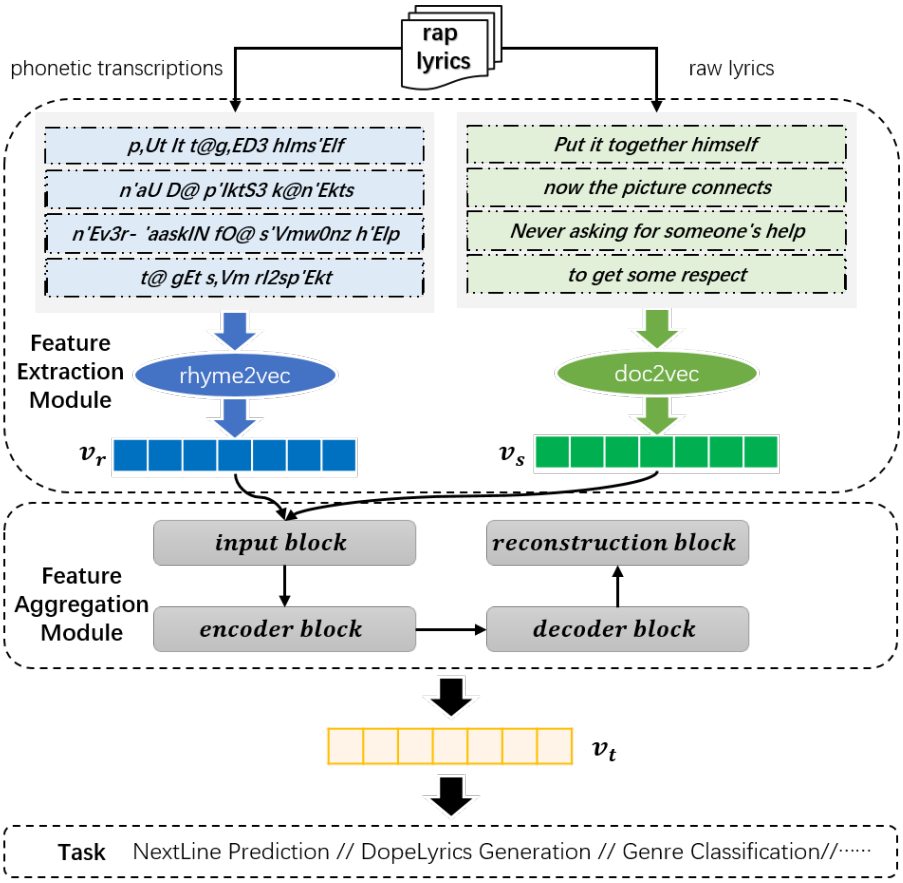

图 4.1 HAVAE的结构

第一节 特征提取模块

特征提取模块（如图 4.1所示）用来获取适当细化的特征。特征提取模块由

两部分组成，分别是韵律部分和语义部分。

4.1.1 韵律部分

本节将介绍一个用于解决问题 1的方法。此方法称为 rhyme2vec。图 4.2展示了由给出的说唱歌词获取韵律表征向量 $\boldsymbol{v_r}$ 的完整过程。

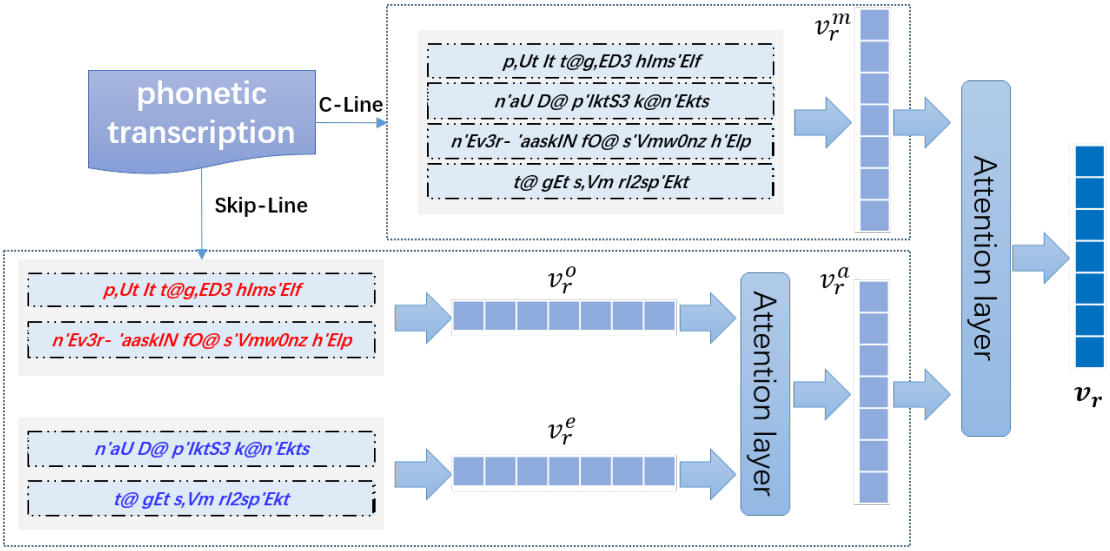

图 4.2 rhyme2vec的结构

每首说唱歌曲包括三个韵律段。首先，将每个音素映射成一个单独的向量，记为 $\boldsymbol{v_p}$。受 [26]学习文本的想法的启发，每个韵律段都被赋予了一个段向量，记作 $\boldsymbol{v_b}$。一个韵律段的向量记为 $\boldsymbol{v_r^b}(\boldsymbol{b} \in \{\boldsymbol{m}, \boldsymbol{o}, \boldsymbol{e}\})$。$\boldsymbol{v_r^b}$ 定义为该韵律段中所有音素向量和段向量的和，即 $\boldsymbol{v_r^b} = \sum \boldsymbol{v}_p + \boldsymbol{v_b}$。利用一个固定长度的滑动窗口将一个韵律段划分为若干个小韵律段，模型通过最大化每个小韵律段中各个音素共同出现在该大韵律段中的概率来训练出所有的 $\boldsymbol{v_p}$ 和 $\boldsymbol{v_b}$。

模型的损失函数写作：

$$\mathcal{L}_{\boldsymbol{r}} = \frac{1}{3N} \times \sum_{i=1}^{3N} \frac{1}{n_i - 2w} \times \sum_{j=w}^{n_i - w} \log P(\boldsymbol{v_p^{j,i}} | (\boldsymbol{v_p^{j-w,i}} : \boldsymbol{v_p^{j+w,i}}, \boldsymbol{v_b^i})), \tag{4.1}$$

其中，n_i 为第 i 个韵律段中音素的个数；w 是滑动窗口一边的大小；$\boldsymbol{v_p^{j,i}}$ 是第 i 个韵律段中第 j 个音素对应的向量；$\boldsymbol{v_b^i}$ 是第 i 个韵律段的段向量。为了提高训练效率和改善训练效果，本文采用了负采样的方法进行训练。目标函数写为：

$$\mathcal{L}_{\boldsymbol{r}} = \mathcal{L}_{r,pos} - k \times \mathcal{L}_{r,neg}, \tag{4.2}$$

其中，k 是每个正样本对应的负样本的个数。

需要注意的是，为了学习单韵模式，本文提出了连续行学习模型记作 C-Line（图 4.2的上半部分）。在 C-Line中，整段输入歌词的韵律形式 $\boldsymbol{L_r^m}$ 作为输入内容，利用上述方法，从对应的音素向量和段向量中计算得到一个分布式向量 $\boldsymbol{v_r^m}$，即为单韵韵律的表示向量，对应于上述方法中的 $\boldsymbol{v_r^b}$。

本文用 $\boldsymbol{v_r^a}$ 指代隔行韵模式的表示向量。由于单韵和隔行韵在一首说唱歌词中的重要性不同，本文引入了注意力层来为 $\boldsymbol{v_r^m}$ 和 $\boldsymbol{v_r^a}$ 赋予不同的权重，以平衡单韵模式和隔行韵模式在最终韵律表示向量中所占权重。注意力层的输入为一个由若干列向量组成的矩阵 $\mathcal{H}$，注意力层的计算方法如下所示：

$$
\begin{aligned}
Attention(\boldsymbol{H}) &= \sum_{j=0}^{m-1} \boldsymbol{\alpha}_j \boldsymbol{h}_j \\
\alpha_i &= \frac{exp\left(\hat{\boldsymbol{h}}_i{}^{\top} \hat{\boldsymbol{h}}_c\right)}{\sum_{j=0}^{m-1} exp\left(\hat{\boldsymbol{h}}_j{}^{\top} \hat{\boldsymbol{h}}_c\right)} \\
\hat{\boldsymbol{h}}_{\boldsymbol{i}} &= tanh(\boldsymbol{W_u} \boldsymbol{h_i} + \boldsymbol{b_u}),
\end{aligned}
\tag{4.3}
$$

其中，m 是输入矩阵 $\mathcal{H}$ 的列数；$\boldsymbol{h_i}$ 是 $\mathcal{H}$ 的第 i 列（$0 \le i \le m-1$）；$\hat{\boldsymbol{h}}_{\boldsymbol{i}}$ 是由 $\boldsymbol{h}_i$ 通过 $tanh$ 函数计算出的一个隐含向量；α_i 是由 $\hat{\boldsymbol{h}}_{\boldsymbol{i}}$ 和一个随机初始化的上下文向量 $\hat{\boldsymbol{h}}_c$ 计算出的权重；$\boldsymbol{W_u}$ 和 $\boldsymbol{b_u}$ 是随机初始化后，在训练中训练出的参数矩阵。最终的融合向量 $\boldsymbol{t}$ 是所有 $\boldsymbol{h_i}$ 的加权和。在融合 $\boldsymbol{v_r^m}$ 和 $\boldsymbol{v_r^a}$ 的注意力层中，$\mathcal{H}$ 由 $\boldsymbol{v_r^m}$ 和 $\boldsymbol{v_r^a}$ 两个列向量组成，即 $\mathcal{H} = [\boldsymbol{v_r^m}; \boldsymbol{v_r^a}]$。$\boldsymbol{v_r} = \boldsymbol{t}$ 是最终的韵律表征向量。

需要注意的是，为了学习单韵模式，本文提出了跳行学习模型记作 Skip-Line（图 4.2的下半部分）。在 Skip-Line中，不同于 C-Line，歌词的韵律形式首先被分为奇数行段（包括所有的奇数行，记作 $\boldsymbol{L_r^o}$）和偶数行段（包括所有的偶数行，记作 $\boldsymbol{L_r^e}$）。$\boldsymbol{v_r^o}$ 和 $\boldsymbol{v_r^e}$ 是以和 $\boldsymbol{v_r^m}$ 同样的方法，分别从 $\boldsymbol{L_r^o}$ 和 $\boldsymbol{L_r^e}$ 计算得到的分布式向量。另外，与融合单韵模式和隔行韵模式类似，本文在 $\boldsymbol{v_r^o}$ 和 $\boldsymbol{v_r^e}$ 之后用了一个注意力层，其中，$\mathcal{H}$ 由 $\boldsymbol{v_r^o}$ 和 $\boldsymbol{v_r^e}$ 两个列向量组成。

最终，$\boldsymbol{L}$ 的韵律的表征向量通过 rhyme2vec方法得到，即

$$
\boldsymbol{v_r} = Att([\boldsymbol{v_r^m}; Attention([\boldsymbol{v_r^o}; \boldsymbol{v_r^e}])]). \tag{4.4}
$$

总的来说，rhyme2vec的算法可以总结为算法 1。

其中，$\textsc{Attention}(a, b)$ 函数即为公式 4.3。

算法 1 rhyme2vec

输入： $\boldsymbol{L_r^m}$, $\boldsymbol{L_r^o}$, $\boldsymbol{L_r^e}$, 窗口大小 w, 向量维度 d, 负采样数量 n
输出： $\boldsymbol{v_r}$

$\boldsymbol{v_r^m} = \text{RHYMEEMBEDDING}(\boldsymbol{L_r^m}, w, d, n)$
$\boldsymbol{v_r^o} = \text{RHYMEEMBEDDING}(\boldsymbol{L_r^o}, w, d, n)$
$\boldsymbol{v_r^e} = \text{RHYMEEMBEDDING}(\boldsymbol{L_r^e}, w, d, n)$
$\boldsymbol{v_r^a} = \text{ATTENTION}([\boldsymbol{v_r^o}, \boldsymbol{v_r^e}])$
$\boldsymbol{v_r} = \text{ATTENTION}([\boldsymbol{v_r^m}, \boldsymbol{v_r^a}])$
return v_r

算法 2 rhymeEmbedding

输入： $\boldsymbol{L_t}$, 窗口大小 w, 向量维度 d, 负采样数量 n
输出： $\boldsymbol{v_t}$

$\boldsymbol{V_t}, \boldsymbol{\theta_t} = \text{INITIALIZEEMBEDDING}(|\boldsymbol{L_t}| + 1, d)$
for each epoch **do**
  for $i = 1 \rightarrow |\boldsymbol{L_t}|$ **do**
    $\boldsymbol{e} = \boldsymbol{0}$
    $\boldsymbol{x_i} = \sum_{k \in \text{CONTEXT}(i)} \boldsymbol{V}_{t,k} + \boldsymbol{V}_{t,0}$
    for $j \in \{i\} \bigcup \text{NEG}(i, n)$ **do**
      $q = \sigma(\boldsymbol{x_i}^\top \boldsymbol{\theta}_{t,j})$
      $g = \eta[T_i(j) - \sigma(\boldsymbol{V}_{t,j}^\top \boldsymbol{x_i})]$
      $\boldsymbol{e} = \boldsymbol{e} + g\boldsymbol{\theta}_{t,j}$
      $\boldsymbol{\theta}_{t,j} = \boldsymbol{\theta}_{t,j} + g\boldsymbol{x_i}$
    end for
    for $j \in \text{CONTEXT}(i) \bigcup \{0\}$ **do**
      $\boldsymbol{V}_{t,j} = \boldsymbol{V}_{t,j} + \boldsymbol{e}$
    end for
  end for
end for
$\boldsymbol{v_t} = \sum_{i=0}^{|\boldsymbol{L_t}|} \boldsymbol{V}_{t,i}$
return $\boldsymbol{v_t}$

算法 2中，INITIALIZEEMBEDDING(a,b) 用于初始化 a 个维度为 b 的向量，以及 a 个维度为 b 的辅助向量；$\boldsymbol{V_t}$ 和 $\boldsymbol{\theta_t}$ 分别为韵律段和其中音素对应的表征向量和辅助向量；CONTEXT(i) 函数的输出为以第 i 个音素为中心、大小为 $2w+1$ 的窗口中除了中心音素以外的音素序号。令训练轮数为 ep，每一轮中，需要以每一个音素为中心，更新窗口中所有音素向量以及该段韵律段的段向量。更新时，需要首先计算更新向量 $\boldsymbol{e}$，这个过程需要计算 $n+1$ 次。所以算法 2的时间复杂度为 $\Theta(ep[9d(n+1)+2(2w+1)d](|\boldsymbol{L_t}|-2w)+d|\boldsymbol{L_t}|)$，即 $\boldsymbol{O}(d\cdot ep\cdot(n+w)(|\boldsymbol{L_t}|-w))$。

4.1.2 语义部分

为了提取语义特征，本文应用了 doc2vec模型 [26]。doc2vec是目前性能最优秀的句嵌入模型之一，分为段落向量的分布式记忆模型（the distributed memory model of paragraph vectors，以下简称 PV-DM）和段落向量的分布式词袋模型（the distributed bag-of-words version of paragraph vectors，以下简称 PV-DBOW）两种模型。实验表明，PV-DM的性能通常优于 PV-DBOW。

本文采用了 PV-DM来学习 $\boldsymbol{L}$ 的语义向量，即 $\boldsymbol{v_s}$。本文选择了负采样的方式来训练 doc2vec模型。在段落的级别上，本文将连续的几行看作一个段落；在歌曲的级别上，我们将整个歌曲看作一个段落。

第二节 特征融合模块

特征融合模块用于解决问题 2，是本文所提出的模型的核心部分。这一部分，本文设计了一个通过融合韵律和语义信息来得出说唱歌词表征向量的VAE网络。此外，模型还引入了注意力机制来平衡韵律部分和语义部分的重要性。特征融合模块的结构如图 4.3所示，其中包括三个主要的阶段，即输入阶段、编码阶段和解码阶段。

4.2.1 输入阶段

在输入阶段，如图 4.3顶部所示，我们用前文提到的注意力层来处理输入数据。通过特征提取模块得到的韵律信息和语义信息的表征向量 $\boldsymbol{v_r}$ 和 $\boldsymbol{v_s}$ 作为输入。注意力层的输出向量 $\boldsymbol{v}$ 将作为后面的阶段的输入。

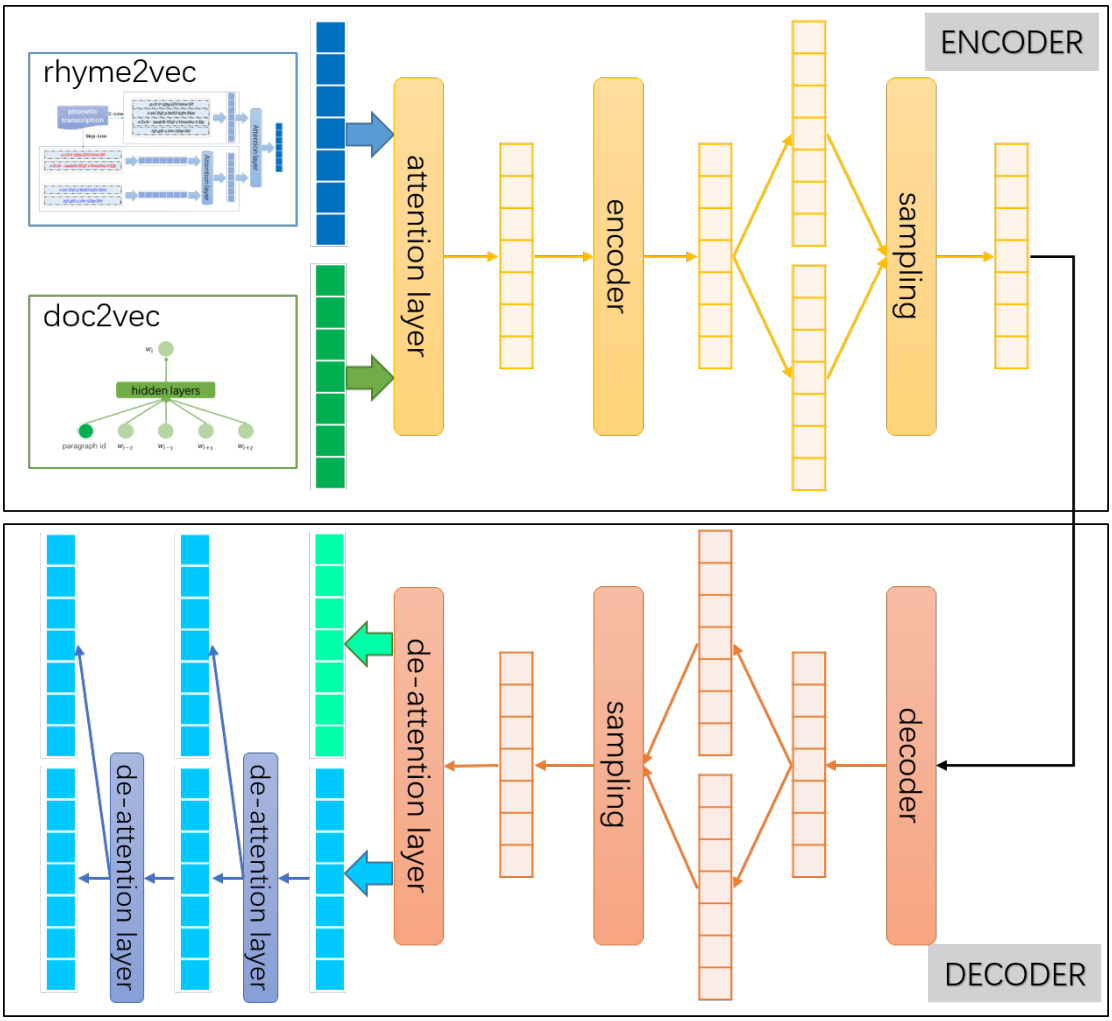

图 4.3 特征融合模块的结构

4.2.2 编码阶段

在编码阶段，首先是若干编码层，每个编码层都是一个全连接层。全连接层的计算公式为 $\boldsymbol{h}_{i+1} = tanh(W_i\boldsymbol{h_i} + b_i)(1 \leq i \leq k)$，其中 k 是编码层的个数，$tanh$ 是每个全连接层的激活函数。将 $\boldsymbol{v}$ 输入全连接层后，得到一个隐含向量 u。VAE网络的目标在于学习隐含向量 $\boldsymbol{z}$，使得 $\boldsymbol{z}$ 可以尽可能多地保留 $\boldsymbol{v}$ 的信息。本文从 $\boldsymbol{u}$ 学习到 $\boldsymbol{z}$ 的期望向量和方差对数向量。令 $\boldsymbol{\mu}_z$ 为 $\boldsymbol{z}$ 的期望向量，$\log \boldsymbol{\sigma}_z^2$ 为 $\boldsymbol{z}$ 的方差对数向量。假设 $\boldsymbol{z}$ 服从 $\mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z^2)$ 的高斯分布，为了网络正常地进行反

向传播，VAE采用了“重参数化”技巧 [21]，采样层的计算过程为：

$$\boldsymbol{z} = \boldsymbol{\mu}_{\boldsymbol{z}} + \boldsymbol{\sigma}_{\boldsymbol{z}} \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{4.5}$$

其中，$\boldsymbol{\varepsilon}$ 是一个从高斯分布中采样得到的随机向量，$\odot$ 表示向量的按位乘法。

4.2.3 解码阶段

解码阶段可以看作是编码阶段的逆运算。隐含向量 $\boldsymbol{z}$ 为解码阶段的输入，被输入若干个解码层，每个解码层都是一个全连接层。最后一个解码层的输出为 $\hat{\boldsymbol{u}}$，$\hat{\boldsymbol{u}}$ 与 $\boldsymbol{u}$ 维度相同。$\hat{\boldsymbol{v}}$ 是通过从 $\mathcal{N}(\boldsymbol{\mu}_{\hat{v}}, \boldsymbol{\sigma}_{\hat{v}}^2 \boldsymbol{I})$ 采样，对 $\boldsymbol{v}$ 进行重构得到的向量。$\boldsymbol{\mu}_{\hat{v}}$ 和 $\log \boldsymbol{\sigma}_{\hat{v}}^2$ 是从 $\hat{\boldsymbol{u}}$ 计算出的 $\hat{\boldsymbol{v}}$ 的期望向量和方差对数向量。

4.2.4 损失函数

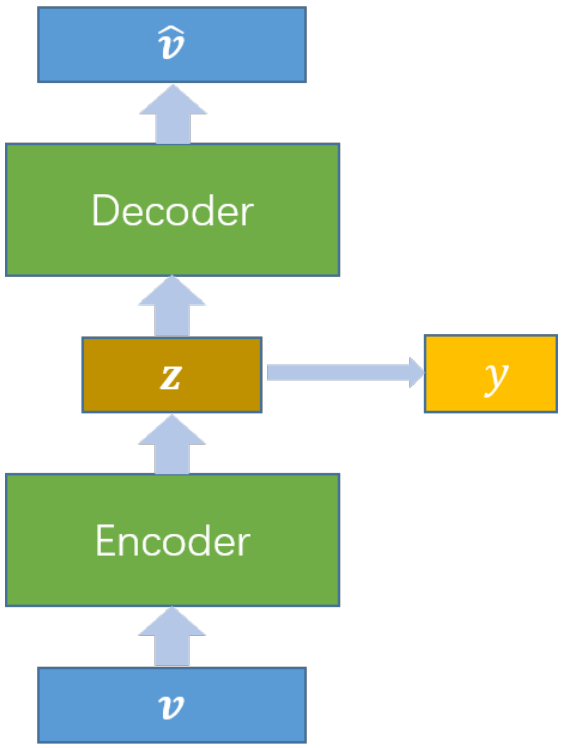

图 4.4 结合 label预测的 VAE模型

受 [35]的启发，HAVAE引入了标签信息。如图 4.4，在图 3.1的基础上加入了对标签的预测，图中右侧的 $\boldsymbol{y}$ 表示在 $\boldsymbol{z}$ 的基础上预测的标签向量，在有可用标签的任务中，加入标签信息可以提高模型的学习效果。所以，HAVAE的损失

函数包括两部分——VAE损失函数和标签损失函数，即

$$\mathcal{L} = \mathcal{L}_{vae} + \alpha \mathcal{L}_{label}. \tag{4.6}$$

4.2.4.1 VAE的损失函数

模型最终学习到的歌词的表征向量为 $\boldsymbol{\mu_z}$，即 $\boldsymbol{v_t} = \boldsymbol{\mu_z}$。VAE的损失函数 $\mathcal{L}_{vae}$ 如下所示：

$$\mathcal{L}_{vae} = -D_{\mathcal{KL}}(\mathcal{Q}(\boldsymbol{z}|\boldsymbol{v})||\mathcal{P}(\boldsymbol{z})) + \log \mathcal{P}(\boldsymbol{v}|\boldsymbol{z}), \tag{4.7}$$

其中，$-D_{\mathcal{KL}}(\mathcal{Q}(\boldsymbol{z}|\boldsymbol{v})||\mathcal{P}(\boldsymbol{z}))$ 是生成损失，$-\log \mathcal{P}(\boldsymbol{v}|\boldsymbol{z})$ 是重构损失。

令 $\mathcal{Q}(\boldsymbol{z}|\boldsymbol{v})$ 为 $\boldsymbol{z}$ 的近似先验分布，$P(\boldsymbol{z})$ 为 $\boldsymbol{z}$ 的先验分布。VAE模型的目标是最小化 $\mathcal{Q}(\boldsymbol{z}|\boldsymbol{v})$ 和 $\mathcal{P}(\boldsymbol{z})$ 之间的差距。此处引入了 $\mathcal{KL}$ 散度（Kullback-Leibler divergence）：

$$D_{\mathcal{KL}}(\mathcal{Q}(\boldsymbol{z}|\boldsymbol{v})||\mathcal{P}(\boldsymbol{z})) = \sum_{i=1}^{N} \mathcal{Q}(\boldsymbol{z_i}|\boldsymbol{v_i}) \times \log[\mathcal{Q}(\boldsymbol{z_i}|\boldsymbol{v_i})/\mathcal{P}(\boldsymbol{z_i})]. \tag{4.8}$$

因为先验分布和后验分布都是高斯分布，等式可以被简化为：

$$D_{\mathcal{KL}}(\mathcal{Q}(\boldsymbol{z}|\boldsymbol{v})||\mathcal{P}(\boldsymbol{z})) = \frac{1}{2}\sum_{i=1}^{N}(1 + \log(\boldsymbol{\sigma_z})_i^2 - (\boldsymbol{\mu_z})_i^2 - (\boldsymbol{\sigma_z})_i^2). \tag{4.9}$$

重构损失部分可以理解为最大化从隐含向量 $\boldsymbol{z}$ 重构出输入向量 $\boldsymbol{v}$ 的概率。由于解码部分是一个多元高斯分布，所以这里本文采用了对数损失函数，因此重构损失为：

$$-\log \mathcal{P}(\boldsymbol{v}|\boldsymbol{z}) = -log\mathcal{N}(\boldsymbol{v}; \boldsymbol{\mu}_{\hat{v}}, \boldsymbol{\sigma}_{\hat{v}}^2 \boldsymbol{I}). \tag{4.10}$$

4.2.4.2 标签损失函数

本文引入了一个标签向量 $\boldsymbol{y}$。$\boldsymbol{y}$ 的维度等于所有样本的类别总数。在HAVAE中，由 $\boldsymbol{v}_t$ 通过 $\hat{\boldsymbol{y}} = F(\boldsymbol{\mu}_z)$ 计算得出对歌词标签的估计 $\hat{\boldsymbol{y}}$，其中 F 是一个带‘sigmoid’激活函数的全连接层。标签损失函数是 $\hat{\boldsymbol{y}}$ 和歌词真实标签 $\boldsymbol{y}$ 之间的差距，本文中标签损失函数为 $\hat{\boldsymbol{y}}$ 和 $\boldsymbol{y}$ 的交叉熵，即：

$$\mathcal{L}_{label} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{j=1}^{M}(y_{i,j}\log Y_{i,j} + (1 - Y_{i,j})\log(1 - y_{i,j})), \tag{4.11}$$

其中 M 是标签集合的大小，即 $\boldsymbol{y}$ 的维度。

第三节 本章小结

本章首先介绍了 HAVAE模型的特征提取模块。对于韵律特征，本文主要关注与英美韵律文学中常见的两种韵律——单韵模式和隔行韵模式。韵律特征的提取应用到了 rhyme2vec模型。rhyme2vec分别先获取了一段韵律文本中与这两种韵律模式相对应的韵律段（$\boldsymbol{L_r^m}$、$\boldsymbol{L_r^o}$ 和 $\boldsymbol{L_r^e}$）的音素符号，然后从这几个韵律段中学习到了相应的韵律向量，再应用注意力机制将这几个韵律段结合到一起，得到整段文本的韵律表征向量 $\boldsymbol{v_r}$。对于语义特征的提取，本文采用了目前效果较好的文本表征学习模型 doc2vec。

之后本章介绍了 HAVAE模型的特征融合模块。特征融合模块是一个结合层次注意力机制的 VAE模型。特征融合模块的输入为特征提取模块输出的韵律表征向量 $\boldsymbol{v_r}$ 和语义表征向量 $\boldsymbol{v_s}$。$\boldsymbol{v_r}$ 和 $\boldsymbol{v_s}$ 在特征融合模块中首先通过注意力机制结合到一起，得到一个整体的表征向量 $\boldsymbol{v}$，之后输入到一个 VAE模型中，经过一个编码过程和解码过程，重构出一个向量 $\hat{\boldsymbol{v}}$，通过最小化 $\boldsymbol{v}$ 和 $\hat{\boldsymbol{v}}$ 之间的差距来优化，最终得到编码出的向量 $\boldsymbol{v_t}$ 作为整段文本的表征向量。

第五章 实验设计与结果分析

第一节 数据集与实验设计

考虑到在线的说唱歌词资源很少，本文采用了从网上爬取的一个说唱歌词语料库[1]。这个语料库包括3154名歌手的65730首歌。

为进行数据清洗，根据[15]，所有标题包含“intro”、“outro”、“skit”、“interlude”和“remi”的歌曲都被从数据集中剔除。在大多数的说唱歌曲中，常见的部分包括包括序曲、副歌、主歌和结尾。在这些部分中，主歌占据主要地位[2]，所以本文的数据集只保留了主歌部分。

第二节 下一行预测

下一行预测任务由[15]提出的。给定一首 n 行的说唱歌曲，假设其前 k $(k < n)$ 行（记作 $\boldsymbol{Q} = \{s_1, s_2, ..., s_k\}$）已知。令 $\boldsymbol{C} = \{c_1, c_2, ..., c_m\}$ 表示 m 行候选歌词。此任务的目的就是从 $\boldsymbol{C}$ 中找到歌词的第 $k+1$ 行，即 s_{k+1}；换句话说，与 s_{k+1} 最相关的歌词 c_i $(1 \leq i \leq m)$ 将被选作匹配目标。

5.2.1 数据集

在此实验中，数据集中的所有说唱歌词都被分成单行的歌词，最终从16697首歌词中分出810567行单独的歌词。整个数据集分为训练集（50%）、测试集（25%）和验证集（25%）。

5.2.2 对比方法

本文选择了以下几种方法作为对比方法：

- **EndRhyme** [15]，该方法通过计算候选歌词行 l_i 和 s_k 结尾匹配元音音素的个数来寻找最适合的下一行；
- **rhyme2vec**，该方法即为 第四章第一节中所描述的韵律表征学习方法；

[1] http://ohhla.com/

[2] https://rappingmanual.com/lesson-rapping-song-structure-learn-how-to-rap/

- **NN5** [15]，该方法是一个字符级别的神经网络。该神经网络将每首歌的最后5行作为查询对象，并将其编码后，寻找最合适的下一行；
- **doc2vec** [26]，该方法为当前最好的文本段落表征学习方法之一，在这个方法中将 $\{s_k; l_i\}$ 作为一个段落；
- **DopeLearning** [15] [3]，该方法是目前最好的说唱歌词表征学习方法。它将一系列的统计特征组合到一起，包括 EndRhyme、EndRhyme-1 [4]、OtherRhyme [5]、LineLength [6]、BOW [7]、LSA [8]以及 NN5 [9]；
- **early fusion** [15, 36]，该方法在多模态融合问题中很常见，所有输入向量都将被拼接在一起作为一个统一的表征向量，即 $\boldsymbol{v_t} = [\boldsymbol{v_r}; \boldsymbol{v_s}]$；
- **EF-AE**，该方法将 $[\boldsymbol{v_r}; \boldsymbol{v_s}]$ 输入一个自编码器（autoencoder，以下简称 AE）网络得到最终的表征向量；
- **EF-VAE**，该方法将 $[\boldsymbol{v_r}; \boldsymbol{v_s}]$ 输入一个 VAE网络得到最终的表征向量。

根据利用到的输入信息，这些方法可以被分为三类，即韵律类、语义类和结合类（结合了韵律和语义信息）。

5.2.3 实验设置

在训练阶段，采用了负采样的方法，即每一个查询行 s_m 都有两个候选歌词行，其中一行是该歌词真正的下一行，作为正例；另一行是从语料库中的其他歌词中随机选择的一行，作为反例。在测试阶段，每一个查询都有一个300行的候选集，其中包含真正的下一行以及299个随机选择的歌词行。

实验将每一个查询与其每一行候选歌词分别结合为一对作为输入。与BOW5和 NN5不同 [10]，本实验中只考虑了查询歌词的最后一行 s_m。α 的值设为1。$\boldsymbol{v_r}$ 和 $\boldsymbol{v_s}$ 的维度设为125。根据 [15]，DopeLearning得到的表征向量最终被输入到 SVM^{rank} [37]工具进行排名。而其他方法得到的表征向量维度都较高，并不适用 SVM^{rank}。本实验设计了一个排名层用于计算每对数据的分数，计算方式为 $score = sigmoid(W\boldsymbol{v_t} + b)$，其中 $score$ $(0-1)$ 是候选歌词行和查询行之间

[3]本文中只采用了其最原始的方法，并进行了复现。由于 NN5方法没有提供源代码，且其结果并没有明显提升，本文中并没有将其加入 DopeLearning方法中。

[4]不同于 EndRhyme，EndRhyme-1统计的是 c_i 和 s_{k-1} 结尾匹配的元音音素的个数。

[5]OtherRhyme统计的是每个单词中匹配元音音素个数的平均数。

[6]LineLength从句子长度的角度计算 c_i 和 s_k 的相似度。

[7]BOW利用 Jaccard距离以及词袋模型计算 s_k 和歌曲最后5行的相似度。

[8]计算 c_i 和 s_k 之间的 LSA相似度。

[9]NN5模型最后 $softmax$ 层的置信度值。

[10]这两种方法在提取语义特征时考虑了查询歌词的最后五行。

的相关性，*score* 越高越好。模型网络中全连接层的激活函数均选用了 *tanh*。模型的优化器选用了 *Adadelta* [38]。模型的训练轮数为20。

表 5.1 本文模型与对照方法的结果对比

Information	Methods	mean rank	MRR	Rec@1	Rec@5	Rec@30	Rec@150
prosodic	EndRhyme	103.2*	0.140*	0.077*	0.181*	0.344*	0.480*
	rhyme2vec	17.7	0.463	0.347	0.592	0.841	0.981
semantic	NN5	84.7*	0.067*	0.020*	0.083*	0.319*	0.793*
	doc2vec	15.5	0.430	0.293	0.588	0.870	0.985
both	DopeLearning	60.8*/79.9	0.243*/0.168	0.169*/0.102	0.304*/0.220	0.527*/0.446	0.855*/0.775
	early fusion	9.6	0.588	0.464	0.738	0.926	0.991
	EF-AE	5.3	0.771	0.683	0.879	0.966	0.995
	EF-VAE	2.3	0.941	0.914	0.973	0.990	0.997
	HAVAE	**1.2**	**0.982**	**0.973**	**0.993**	**0.999**	**1.000**

带*的结果是 [15]中展示的结果，其他结果都是根据本文中设置重新运行得到的结果。

5.2.4 结果评估与分析

实验结果由 mean rank、mean reciprocal rank（以下简称 MRR）和 Rec@n（$n=1,5,30,150$）三个标准评估进行评估。令 Q 表示查询行序列，r_i 表示在第 i 个查询中正确的下一行在候选歌词中的排名。mean rank是 r_i 的平均值，范围在 $[1,300]$，越小越好。MRR是 $\frac{1}{r_i}$ 的平均值，越大越好。Rec@n是 $r_i \leq n$ 的频率，值越大越好。

实验结果如表 5.1所示。显然，本文提出的方法比其他对比方法的性能都要突出，达到了目前最高的水平。从结果中，主要有如下发现：

5.2.4.1 HAVAE的有效性

在所有的这些方法中，本文提出的模型取得了最好的效果。mean rank值为**1.2**，极为接近1，远优于其余方法中最好的方法。DopeLearning的原始论文中，mean rank为60.8，而本文复现的结果为79.9，二者皆高于60。在 MRR方面，本文模型的方法结果为0.982，优于 DopeLearning的0.168和0.243。在 Rec@n上的性能展示了相似的情况——本文模型得到了远好于其他方法的结果。总的来说，本文的模型比当前最好的方法表现更好，表明本文提出的模型是有效的。造成这种差距的主要原因是提取了更有效的韵律特征和语义特征（分布式表征学习而非简单的统计学特征），并进行了适当的融合（基于 VAE的框架）。

5.2.4.2 rhyme2vec的有效性

与EndRhyme相比，rhyme2vec很明显在实验中的几个评价标准（mean rank、MRR和 Rec@k）上都得到了比其他方法更好的结果。这种结果表示本文提出的方法 rhyme2vec学习到的韵律表征向量，通过考虑多种韵律模式，得到了比其他方法得到的韵律向量更为有效。

5.2.4.3 韵律信息对整体表征向量的提升

由表格中数据可见，单独利用语义表征（doc2vec）或韵律表征（rhyme2vec）的结果都较差（mean rank大于15，MRR小于0.5）。而将两种特征融合后，最终结果得到了显著提升。同样的情况在 DopeLearning中也得到了体现，DopeLearning的结果也要好于单独应用 EndRhyme或 NN5特征。这两个结果同时证明了同时利用韵律特征和语义特征可以得到更好的效果。

5.2.4.4 特征融合模型的有效性

early fusion方法将所有特征的表征向量进行拼接，得到的向量作为最终的融合向量（与 DopeLearning相同的方法），最终在 mean rank和 MRR上分别得到了9.6和0.588的结果。与 early fusion相比，可以看到 EF-AE得到了更好的结果，其原因有可能是因为 EF-AE加入了自编码器网络，学到了不同特征之间的相互联系 [39]。另外，EF-VAE的表现比 EF-AE更好，这也表明 VAE在这个歌词学习的任务上比 AE模型更为有效。而 HAVAE模型比 EF-VAE的结果更好，表明注意力机制的有效性，其结果的提升是因为注意力机制强化了语义和韵律信息之间的练习，提升了模型的效果。

5.2.4.5 其他发现

实验结果显示，doc2vec在 mean rank上的结果优于 rhyme2vec。相似的，NN5网络的结果在 mean rank上同样优于 EndRhyme。这有可能是由于在歌词中，语义信息比韵律信息更为重要。然而从 MRR和 Rec@n的结果来看，rhyme2vec在 MRR、Rec@1、Rec@5上的结果比 doc2vec更好，这说明 rhyme2vec得到的结果更为极端 doc2vec，即预测的结果中给更多的正确结果预测了一个很高或很低排名，而 doc2vec则会得到一个更为中庸的结果，给更多的正确结果预测到了一个中上的位置。

表 5.2 不同韵律模式的性能

韵律模式	mean rank	MRR
C-Line	60.1030	0.1542
Skip-Line	61.3070	0.1498
sum	24.0840	0.4067
rhyme2vec	**17.0810**	**0.4714**

为探索不同韵律模式对结果的贡献，本文还分别对 C-Line韵律向量、Skip-Line韵律向量和两种向量的和向量 sum进行了实验，实验结果如表 5.2所示。由结果可知，C-Line的结果略微好于 Skip-Line，但差距并不明显，说明在说唱歌词中，单韵模式和隔行韵模式的重要性基本相同；而 sum的结果优于 C-Line和Skip-Line说明同时学习单韵模式和隔行韵模式可以明显提升模型的性能，两种韵律模式的结合是有价值的；rhyme2vec的结果好于 sum，也再一次证明了注意力机制的有效性。

5.2.5 参数讨论

为优化模型效果，本文还对模型进行了参数讨论，包括公式 4.6中的 α、语义特征模块输入行数 k 以及模型在每轮的实验结果和损失函数。

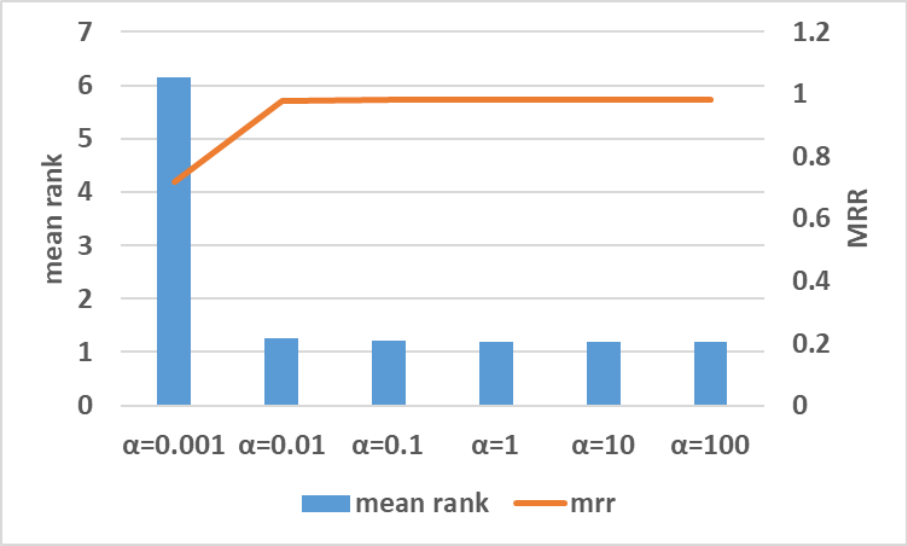

图 5.1 α 对结果的影响

对参数 α 的讨论结果如图 5.1所示，实验只是对 α 的数量级进行了讨论，这是因为，在同一数量级，α 对于实验结果的影响并不明显。α 的数量级从 10^{-3} 到 10^{-2} 时，效果大幅提升；当 $\alpha \geq 10^{-2}$ 时，模型的效果几乎没有变化，mean rank约为 1.2，MRR约为 0.98。这表明加入标签信息对于模型的训练有所提升，但模型并不依赖标签信息，而是更偏重与 VAE损失函数，即更需要表征向量尽可能多地保存原数据的信息。

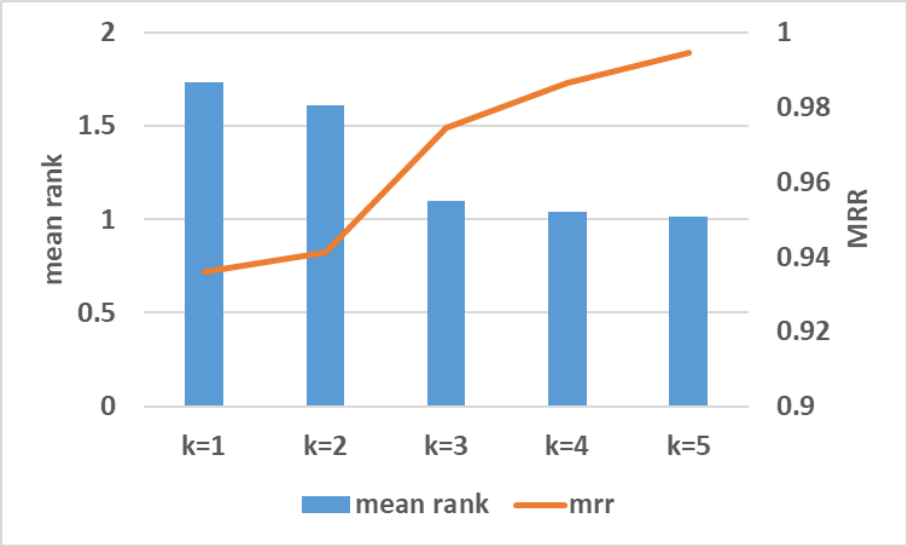

图 5.2 k 对结果的影响

对参数 k 的讨论结果如图 5.2所示，这里为了更明显地体现 k 的影响，没有用 HAVAE模型进行参数讨论，而是单纯用语义特征提取模块 doc2vec模型进行了参数讨论。从图中可见，随着 k 的提升，实验效果明显提升，说明更多的查询行更有利于下一行预测的实验效果。但是由于提高 k 的时间开销太大，所以最终实验时只选取了 $k=1$。

本文还对模型训练过程中 mean rank分数、MRR分数和损失函数的变化进行了记录，结果如图 5.3所示，(a)为模型的 mean rank得分，(b)为模型的 MRR得分，(c)为模型的标签损失函数 $\mathcal{L}_{label}$，(d)为模型的 VAE损失函数 $\mathcal{L}_{vae}$，每幅图的横轴为模型训练的轮数。由图 5.3(a)可知，模型在训练到第3-4轮时，mean rank分数就已达到1.2以下；由图 5.3(b)可知，模型在训练到第四轮时，MRR分数达到了最高值；由图 5.3(c)可知，模型在训练到第4-5轮时，$\mathcal{L}_{label}$ 达到了最低值；由图 5.3(d)可知，模型的 $\mathcal{L}_{vae}$ 虽然一直在下降，但在第3-4轮后便趋于平稳。

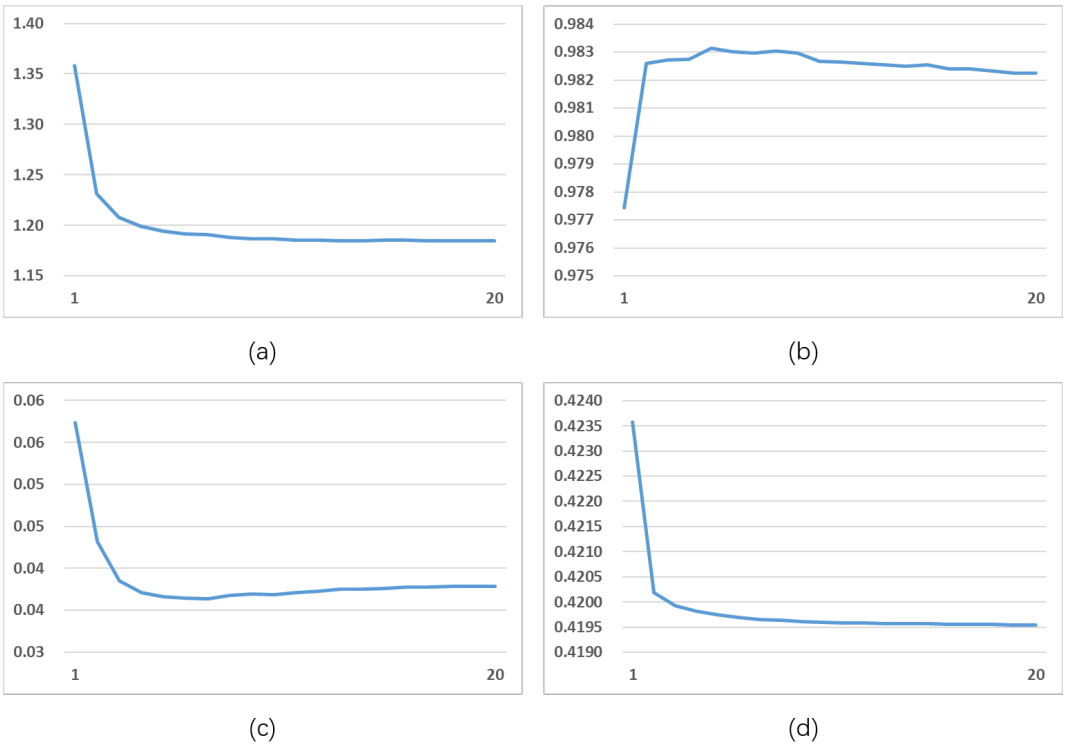

图 5.3 模型评价结果和损失函数随训练过程的变化

由以上发现可知模型收敛速度很快，分析有可能是由于数据量太小导致的。

第三节 说唱歌曲生成

基于下一行预测实验，本文进行了另一个实验，即说唱歌词生成。该实验给定一系列单行的说唱歌词 $\boldsymbol{L}=\{l_1,l_2,...,l_n\}$，以及一个候选歌词集合 $\boldsymbol{C}=c_1,c_2,...,c_m$，其中，$n$ 为集合 $\boldsymbol{L}$ 的大小，m 为集合 $\boldsymbol{C}$ 的大小。对于 $\boldsymbol{L}$ 中的每行歌词 $l_i\ (1\leq i\leq n)$，模型需要从 l_i 出发，利用候选歌词生成一段 t 行的说唱歌词。实验的对比方法选择了 DopeLearning [15]。

5.3.1 数据集

实验所用的 $\boldsymbol{L}$ 是从 5.2.1所述数据集中随机挑选的100行歌词，$\boldsymbol{C}$ 中包含810567行歌词。

5.3.2 实验设置

令 $T_{i,j}$ 为从 l_i 出发，生成的 j 行的歌词段。模型要从初始歌词 l_i（亦即 $T_{i,1}$）开始，生成一段 t 行的歌词，需要经过 $t-1$ 次预测，第 j 次预测的输入为 $T_{i,j}$，模型得到输入后，将从 $\boldsymbol{C}$ 中挑选出最有可能的下一行歌词 c，将 c 接在 $T_{i,j}$ 的末尾，得到 $j+1$ 行的歌词段 $T_{i,j+1}$ 作为第 $j+1$ 次预测的输入。为避免生成歌曲中出现重复歌词，实验中每一步都会将所选作为下一行的歌词从 $\boldsymbol{C}$ 中剔除。本文将最终生成歌词的长度 t 设置为 16。实验中的模型均为通过 5.2.1所述下一行实验的数据集训练得到的模型。

5.3.3 结果评估及分析

5.3.3.1 评价标准

好的说唱歌词的应该通顺、流畅，主题应该明确，并且韵律应该足够有吸引力。然而，歌词的流畅度、主题的明确程度并没有一个量化的评价方法，而完全靠人工评估并不客观，且需要大量的人力与经济成本。因此，本文只采用了对韵律的量化评价方法来评价模型生成的说唱歌曲。

本文采用了两种韵律密度评分，称为 RD2010和 RD2016。RD2010由 [18]提出，该评价标准将一首歌词中押韵的音节在所有音节中所占比例作为对该首歌词的韵律密度评分，一首歌词的 RD2010分数越高，说明该歌词押韵的语句越多。RD2016由 [15]提出，该评价标准通过计算一首歌词中相邻行单词的最长匹配元音序列的平均值来计算该首歌词的韵律密度分数，一首歌词的 RD2016分数越高，说明该首歌词相邻行单词之间押韵越多。

5.3.4 实验结果分析

实验结果如图 5.4和图 5.5所示。由图 5.4和图 5.5可见，HAVAE和 DopeLearning在 RD2010上的整体得分情况比较接近；而在 RD2016上，HAVAE的得分明显高于 DopeLearning，仅在个别歌曲中，HAVAE的得分低于 DopeLearning。由以上结果可知，HAVAE所生成歌词的韵律比 DopeLearning所生成歌词更好，同 5.2.4得到了相同的结论。

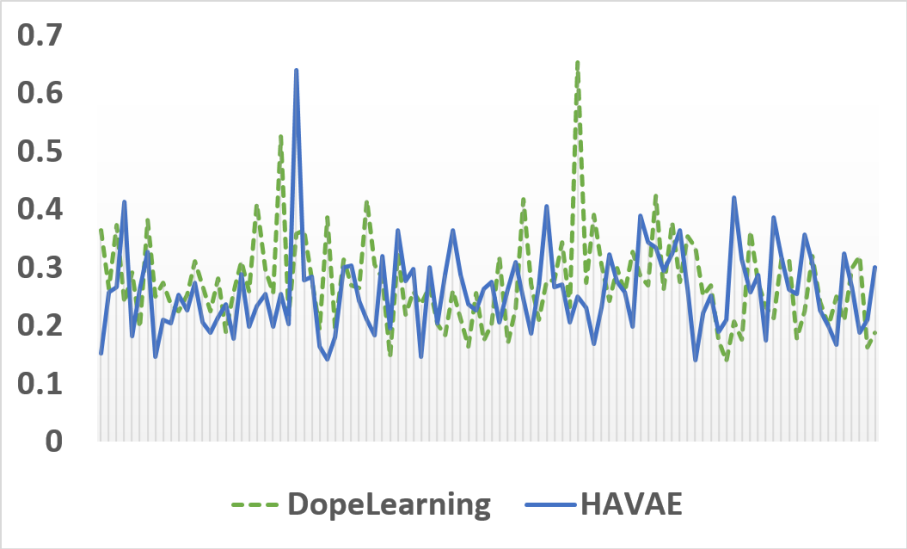

图 5.4 RD2010分数

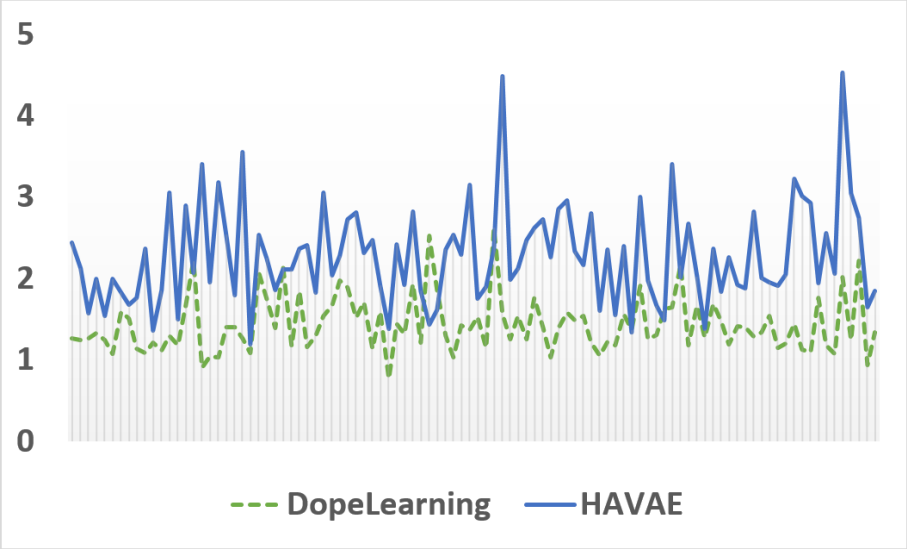

图 5.5 RD2016分数

第四节 说唱歌词流派分类

为展示 HAVAE和 rhyme2vec的泛化能力，本文设计了歌曲级别的说唱流派多标签分类任务作为补充。给定一个说唱歌曲的集合 $\boldsymbol{S}=\{s_1,s_2,...,s_N\}$ 和一个说唱歌词的标签集合 $\boldsymbol{G}=\{g_1,g_2,...,g_M\}$，要求预测出每一首歌的标签集合，即每首歌对应多个标签，其中 N 是集合 $\boldsymbol{S}$ 的大小，M 是集合 $\boldsymbol{G}$ 的大小。也就是

说，该任务是一个多标签分类任务。

5.4.1 数据集

本文采用了一个包含10167首歌曲和9种流派[11]的流派分类数据集，即 $\boldsymbol{G}$ 的大小 $M=9$，数据集中 9 种标签的数量如表 5.3所示。数据集中每首歌词都对应一名歌手，歌手则对应了一个标签集合 $\boldsymbol{G_i}$，该首歌词的标签集合即为该名歌手对应的标签集合。数据集分为训练集和测试集，训练集包含9150首歌曲，测试集包含1017首歌曲。

表 5.3 数据集中流派标签数量分布

0	1	2	3	4	5	6	7	8
638	336	3507	268	1178	793	1192	3680	2375

5.4.2 对比方法

- **RhymeAPP** [25]，该方法是一个计算说唱歌词统计特征的歌词分析工具；
- **rhyme2vec**，该方法即为 第四章第一节中所描述的韵律表征学习方法；
- **doc2vec** [26]，该方法同 5.2.2，将整首歌词作为一个段落进行学习；
- **HAN-L** [40]，该方法是当前最优秀的歌词流派分类方法，采用了循环神经网络（recurrent neural network，即 RNN），并在词和句子层面分别应用了注意力机制，该方法仅学习了语义特征。

5.4.3 实验设置

本实验将歌词输入到各个模型中，得到相应的特征向量，然后将这些特征向量输入一个分类网络中，该网络为每一个输入样本计算出一个 9 维的向量 $\boldsymbol{y}$，$\boldsymbol{y}$ 的每一个元素对应一个流派，表示一首歌属于该流派的可能性。该分类网络有一个参数，即阈值 t，用于决定一首歌是否属于某个流派，此过程如下所示：

$$\boldsymbol{y_i'} = \begin{cases} 0, & if\ \boldsymbol{y_i} \leq t \\ 1, & if\ \boldsymbol{y_i} > t \end{cases} \tag{5.1}$$

最终 $\boldsymbol{y_i'}$ 为分类结果。

[11]这9种流派包括 Alternate、Christian、East Coast、Grime、Hardcore、Horrorcore、Midwest、Southern和 West Coast。

5.4.4 结果评估及分析

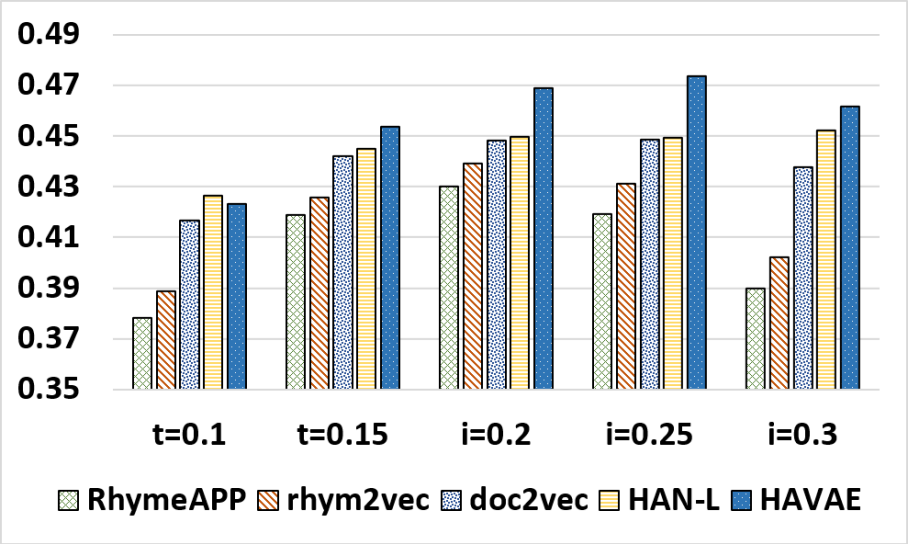

图 5.6 micro-f1分数

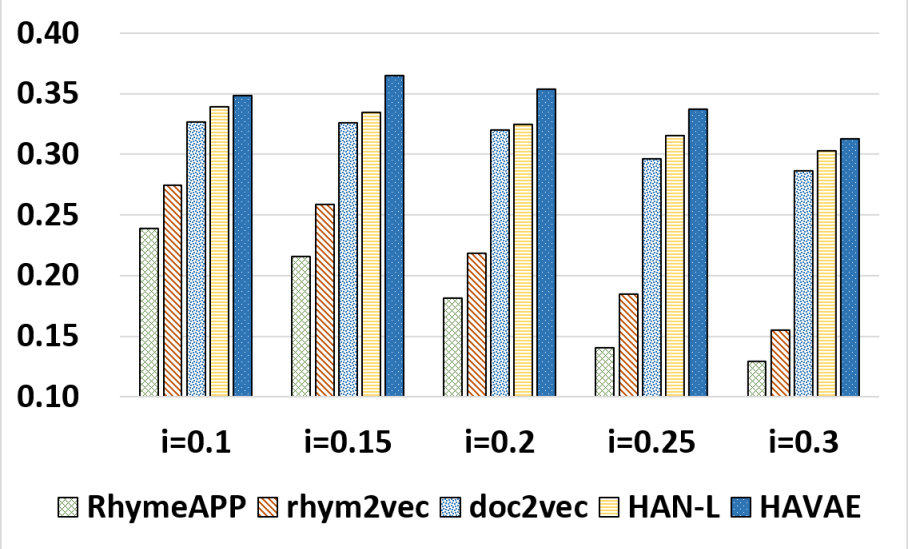

图 5.7 macro-f1分数

本文用 micro-f1和 macro-f1两个标准来评估最终的分类结果。f1是一种用来结合 precision和 recall评估结果的评价标准，micro-f1和 macro-f1是对 f1不同的平均方式。

micro-f1的计算方式如下：

$$\begin{aligned} precision_{micro} &= \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m}(TP_i + FP_i)} \\ recall_{micro} &= \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m}(TP_i + FN_i)} \\ micro - f1 &= \frac{(\beta^2+1) \times precision_{micro} \times recall_{micro}}{\beta^2 \times precision_{micro} + recall_{micro}} \end{aligned} \tag{5.2}$$

其中 TP_i 为第 i 类中 true positive的数量，FP_i 为第 i 类中的 false positive的数量，FN_i 为第 i 类中的 false negative的数量。

macro-f1的计算方式为：

$$\begin{aligned} precision_i &= \frac{TP_i}{TP_i + FP_i} \\ recall_i &= \frac{TP_i}{TP_i + FN_i} \\ f1_i &= \frac{(\beta^2+1) \times precision_i \times recall_i}{\beta^2 \times precision_i + recall_i} \\ micro - f1 &= \frac{\sum_{i=1}^{m} f1_i}{m} \end{aligned} \tag{5.3}$$

其中 TP_i、FP_i 和 FN_i 的含义与公式 5.2中相同。

最终的分类评估结果如图 5.6和图 5.7所示。总的来说，在大部分情况下，HAVAE比其他方法的效果更好，尤其是当 $t = 0.2$ 的时候。从结果中，主要有以下发现：

5.4.4.1 HAVAE在歌曲级别的有效性

与下一行预测任务不同，歌词流派分类任务的数据是在歌曲级别上。而从图 5.6和图 5.7中可以看到，HAVAE仍然比其他方法更为有效。由此可见，输入数据中单个样本的长短并不会对 HAVAE的效果造成很大影响，此结果可以在一定程度上说明 HAVAE的泛化能力较好。

5.4.4.2 语义与韵律特征的重要性

由表 5.1和 5.2.4.5可知，在下一行预测任务中，doc2vec和 rhyme2vec的结果相差并不多，然而，在歌词流派分类中，doc2vec的结果却远优于 rhyme2vec的结果，出现这种结果，很有可能是因为，对于歌词中的几行，韵律的模式更为明显，很有可能只有一种韵律模式；而对于整首歌词，可能会有多种韵律模式

的混合，韵律特征的效果会受到一定的影响。而语义特征并不会受到这种影响，反而有可能因为短文本中包含的语义不完整而导致整首歌词的学习效果相对于部分歌词的学习效果有所提升。当然，不排除 doc2vec模型比 rhyme2vec模型更适用于多标签分类任务。

第五节 本章小结

本章通过三个实验从不同的角度证明了 HAVAE模型的有效性。

首先是下一行预测实验，这个实验给定一组文本段作为查询集，一组文本行作为候选集，实验的模型要根据每一个查询段进行检索，将候选行按出现在查询段下一行的可能性排序，最有可能出现在下一行的排在最前。最终，在 mean rank评价标准上，HAVAE模型比表现最好的对比方法 DopeLearning的效果高出一个数量级；在 MRR评价标准上，HAVAE比 DopeLearning的效果高3倍。另外，实验还通过分别测试不同韵律模式模型和 rhyme2vec模型，证明了 rhyme2vec模型中注意力机制的有效性。

之后是歌曲生成实验，该实验建立在在下一行预测实验的基础上。实验的输入为一组单行的歌词，以及一个候选单行歌词集。模型需要以每一个输入的单行歌词为现有文本，从候选的单行歌词集中重复的挑选下一行，与现有文本组成新的输入，直到生成所需长度的歌词。实验的评价标准为 RD2010和 RD2016。在 RD2010上，HAVAE的效果和对比方法 DopeLearning的表现接近，而在 RD2016上，HAVAE的效果则明显高于 DopeLearning。下一行预测实验和歌曲生成实验证明了 HAVAE模型在韵律文本检索任务上的有效性。

最后，本章介绍了歌词流派分类实验。该实验的数据包括一个歌词集合，以及集合中每首歌词对应的流派标签。模型通过分析每一首歌词，对其进行流派分类。最终，在 macro-F1和 micro-F1两个评价标准上，HAVAE都明显高于其他对比方法。该实验证明了 HAVAE模型在韵律文本分类任务上的有效性。

第六章 结论与展望

韵律文本是一种特殊类型的文本，相对于通俗的自然语言，韵律文本分析与学习值得进行专门的研究。然而目前韵律文本分析与学习存在以下一些问题：韵律文本分析与学习的数据是非结构化的，相对于结构化数据，非结构化数据中难以获取计算机需要的形式语义，因此通过计算机进行处理和分析有很大难度；且相对于自然语言，韵律文本中还添加了韵律和形式的特征，而现有的NLP技术由于没有考虑韵律和结构的特征，并不适合直接应用于韵律的分析；不同类型的韵律文本的结构和韵律特征差异较大，一种算法难以广泛应用于大部分的韵律文本学习任务;；许多韵律文本学习方法并没有充分利用韵律文本的特征，许多都只是单纯利用了韵律文本的语义特征或韵律特征，没有把二者结合分析；许多韵律文本学习方法是针对特定类型的韵律文本进行的，泛用性很差。

为解决以上问题，本文提出了 rhyme2vec和 HAVAE两个模型。rhyme2vec用来学习韵律表征向量。这个方法包含两个模型，即连续行韵律和隔行韵律。通过整合这两个模型，rhyme2vec可以很好地处理韵律模式的多种特征。HAVAE用于融合韵律文本的韵律特征和语义特征。该框架旨在处理韵律文本的表征学习问题，利用注意力机制对韵律信息进行了有效整合且对语义与韵律信息进行了无缝整合。

最终，本文通过实验验证了 rhyme2vec和 HAVAE训练得到的表征向量在检索和分类等任务上的有效性，实验包括下一行预测、流派分类、歌词生成。通过与现有的一些表现较好的韵律文本学习方法进行比较，最终结果显示 rhyme2vec和 HAVAE相对于这些方法更为有效。在下一行预测实验中，HAVAE模型在 mean rank评价标准上比表现最好的对比方法 DopeLearning的效果高出一个数量级；在 MRR评价标准上比 DopeLearning的效果高3倍。另外，实验还通过分别测试不同韵律模式模型和 rhyme2vec模型，证明了 rhyme2vec模型中注意力机制的有效性。在歌曲生成实验中，HAVAE在 RD2010上的效果和对比方法 DopeLearning的表现接近，而在 RD2016上的效果则明显高于 DopeLearning。下一行预测实验和歌曲生成实验证明了 HAVAE模型在韵律文

本检索任务上的有效性。在歌词流派分类实验中，HAVAE在 macro-F1和 micro-F1两个评价标准上都明显高于其他对比方法，该实验证明了 HAVAE模型在韵律文本分类任务上的有效性。

HAVAE的作用只在于学习一段韵律文本的表征向量，其作用主要在于对整段韵律文本的分析与学习，如对韵律文本的分类、聚类、检索等。这些任务在实际应用中的并不常用，而更为常见的是韵律文本的生成问题。在日后的研究中，韵律文本的生成问题将是一个更加值得研究的课题。

参考文献

[1] 肖曼琼. “陌生化”: 从诗歌创作到诗歌翻译. 外语教学, 2008, 29(2): 93 ∼ 96.

[2] 王建刚. 日本俳句与中国六朝美学. 美育学刊, 2016, (2016 年 02): 9 ∼ 18.

[3] 宋园方. 诗词平仄与韵律判定的实现. 科技信息, 2009, (10): 112 ∼ 112.

[4] 赵广竹. 英美诗歌中韵律节奏的音乐性. 鞍山师范学院学报, 2007, (1): 70 ∼ 73.

[5] Oliveira H G. PoeTryMe: a versatile platform for poetry generation. Computational Creativity, Concept Invention, and General Intelligence, 2012, 1: 21.

[6] Kurzweil R. Ray kurzweil’s cybernetic poet, 2001.

[7] 蒋锐滢, 崔磊, 何晶, et al. 基于主题模型和统计机器翻译方法的中文格律诗自动生成. 计算机学报, 2015, 38(12): 2426 ∼ 2436.

[8] He J, Zhou M and Jiang L. Generating chinese classical poems with statistical machine translation models. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence, 2012: 1650 ∼ 1656.

[9] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association, 2010.

[10] Zhang X, Lapata M. Chinese Poetry Generation with Recurrent Neural Networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014: 670 ∼ 680.

[11] Jamal N, Mohd M and Noah S A. Poetry classification using support vector machines. Journal of Computer Science, 2012, 8(9): 1441.

[12] Lou A, Inkpen D and Tanasescu C. Multilabel Subject-Based Classification of Poetry. In: FLAIRS Conference, 2015: 187 ∼ 192.

[13] Mauch M, MacCallum R M, Levy M, et al. The evolution of popular music: USA 1960–2010. Royal Society open science, 2015, 2(5): 150081.

[14] Potash P, Romanov A and Rumshisky A. GhostWriter: Using an LSTM for Automatic Rap Lyric Generation. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, 2015: 165 ∼ 177.

[15] Malmi E, Takala P, Toivonen H, et al. DopeLearning: A Computational Approach to Rap Lyrics Generation. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, California, USA, 2016: 195 ∼ 204.

[16] Hirjee H, Brown D G. Automatic Detection of Internal and Imperfect Rhymes in Rap Lyrics. In: Proceedings of the 10th International Society for Music Information Retrieval Conference, 2009: 711 ∼ 716.

[17] Addanki K, Wu D. Unsupervised rhyme scheme identification in hip hop lyrics using hidden Markov models. In: Proceeding of International Conference on Statistical Language and Speech Processing, 2013: 39 $\sim$ 50.

[18] Hirjee H, Brown D G. Using Automated Rhyme Detection to Characterize Rhyming Style in Rap Music. Empirical Musicology Review, 2010, 5(4).

[19] Wu D, Addanki V S K, Saers M S, et al. Learning to freestyle: Hip hop challenge-response induction via transduction rule segmentation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, United States, 2013: 102 $\sim$ 112.

[20] Potash P, Romanov A and Rumshisky A. Evaluating Creative Language Generation: The Case of Rap Lyric Ghostwriting. arXiv preprint arXiv:1612.03205, 2016.

[21] Kingma D P, Welling M. Auto-Encoding Variational Bayes. In: Proceedings of the 2nd International Conference on Learning Representations, 2014.

[22] Hou X, Shen L, Sun K, et al. Deep feature consistent variational autoencoder. In: Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, 2017: 1133 $\sim$ 1141.

[23] Alexey T, Ivan P Y. Music generation with variational recurrent autoencoder supported by history. arXiv preprint arXiv:1705.05458, 2017.

[24] Hadjeres G, Nielsen F and Pachet F. GLSR-VAE: Geodesic Latent Space Regularization for Variational AutoEncoder Architectures. arXiv preprint arXiv:1707.04588, 2017.

[25] Hirjee H, Brown D G. Rhyme analyzer: An analysis tool for rap lyrics. In: Proceedings of the 11th International Society for Music Information Retrieval Conference, 2010.

[26] Quoc V L, Tomas M. Distributed Representations of Sentences and Documents. In: Proceedings of the 31st International Conference on Machine Learning. Beijing, China, 2014: 1188 $\sim$ 1196.

[27] Hu X, Downie J S and Ehmann A F. Lyric text mining in music mood classification. American music, 2009, 183(5,049): 2 $\sim$ 209.

[28] He H, Jin J, Xiong Y, et al. Language feature mining for music emotion classification via supervised learning from lyrics. In: International Symposium on Intelligence Computation and Applications, 2008: 426 $\sim$ 435.

[29] Cai L, Gao H and Ji S. Multi-Stage Variational Auto-Encoders for Coarse-to-Fine Image Generation. arXiv preprint arXiv:1705.07202, 2017.

[30] Semeniuta S, Severyn A and Barth E. A Hybrid Convolutional Variational Autoencoder for Text Generation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017: 627 $\sim$ 637.

[31] Fabius O, Amersfoort J R van. Variational recurrent auto-encoders. arXiv preprint arXiv:1412.6581, 2014.

[32] Xu W, Sun H, Deng C, et al. Variational Autoencoder for Semi-Supervised Text Classification. In: AAAI, 2017: 3358 $\sim$ 3364.

[33] Morin F, Bengio Y. Hierarchical Probabilistic Neural Network Language Model. In: Aistats, 2005: 246 $\sim$ 252.

[34] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, 2013: 3111 $\sim$ 3119.

[35] Kingma D P, Rezende D J, Mohamed S, et al. Semi-Supervised Learning with Deep Generative Models. Advances in Neural Information Processing Systems, 2014, 4: 3581 $\sim$ 3589.

[36] Chen X, Wang Y and Liu Q. Visual and Textual Sentiment Analysis Using Deep Fusion Convolutional Neural Networks. In: Proceedings of the 2017 IEEE International Conference on Image Processing, 2017.

[37] Thorsten J. Training linear SVMs in linear time. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006: 217 $\sim$ 226.

[38] Zeiler M D. ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701, 2012.

[39] Bengio Y. Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2009, 2(1): 1 $\sim$ 127.

[40] Tsaptsinos A. Lyrics-Based Music Genre Classification Using a Hierarchical Attention Network. In: Proceedings of the 18th International Society for Music Information Retrieval Conference, 2017.