基于改进 TextRank 的藏文抽取式摘要生成

李维[1,2]，闫晓东[1,2]，解晓庆[1,2]

（1.中央民族大学 信息工程学院，北京 100081；

2.中央民族大学 国家语言资源监测与研究中心 少数民族语言分中心，北京 100081）

摘要：目前，藏文抽取式文本摘要方法主要是提取文本自身的特征，对句子进行打分，不能挖掘句子中深层的语义信息。本文提出了一种改进的藏文抽取式摘要生成方法。此方法将外部语料库的信息以词向量的形式融入到 TextRank 算法，通过 TextRank 与词向量的结合，把句子中每个词语映射到高维词库形成句向量，进行迭代为句子打分，并选取分值最高的句子重新排序作为文本的摘要。实验结果表明该方法能有效提升摘要质量。本文还在传统 Rouge 评测方法的基础上，提出了一种采用句子语义相似度计算的方式进行摘要评测的方法。

关键词：文本摘要；TextRank；词向量；句子相似度

中图分类号：TP391　　**文献标识码：**A

Tibetan abstract generation based on improved TextRank

LI Wei[1,2]，YAN Xiaodong[1,2]，XIE Xiaoqing[1,2]

(1.School of Information Engineering, Minzu University of China, Beijing 100081, China;

2.Minority Languages Branch, National Language Resource and Monitoring Research Center,

Minzu University of China, Beijing 100081, China)

Abstract: At present, Instead of capturing the deep semantic information in the sentence, Tibetan text abstraction method mainly extracts features of text itself and make score of every sentence. In this paper, we propose an improved method for Tibetan extractive summarization. This method integrates the information of the external corpus into the TextRank algorithm in the form of word vector. Through the combination of TextRank and word vector, each word in the sentence is mapped to a sentence vector by the high-dimensional lexicon, and iteratively scores the sentence. We select the sentences with the highest score and reorder them as a summary of the text. The experimental results demonstrate that the method can effectively improve the quality of the abstract. Based on the traditional Rouge evaluation method, we propose a method for abstract evaluation based on sentence semantic similarity.

Key words: Text summarization; TextRank; Word vector; Sentence similarity;

0 引言

自 1997 年国际标准化组织在决议中通过藏文编码方案以来，藏文的互联网产业蓬勃发展。互联网中藏文信息数量的激增导致藏文面临着信息过载的问题。自动文摘（automatic summarization/abstracting）为解决这一问题应运而生。自动文摘是利用计算机自动实现文本分析，内容归纳和摘要自动生成的技术[1]。根据摘要和原文的关系进行划分，可以将文本摘要分为抽取式（extract）和理解式（abstract）两种[2]，前者通过提取原文的句子形成摘要，后者是以人工智能，特别是自然语言理解为基础发展起来的方法，其通过对原文在语义上深层次的理解，重新对文本进行表述[3]。但人类语言包括字、词、短语、句子、段落、文档、多文档几个级别，理解难度依次递增，所以摘录式摘要通常好于理解式摘要[4-6]。

1958 年，IBM 公司的 Luhn 基于高频词语的评分提出了一种文本摘要方法[7]，开启了自动文本摘要研究的先河。到目前为止，大多数的研究都集中在抽取式摘要当中，抽取式文摘中心思想是从文章中找出能概括文章主题思想的句子作为摘要，这些句子称为关键句，而关键句选取通常通过分析文本的词频、标题、位置、句法结构、

线索词、指示性短语等。抽取式算法大致分为四个类别：（1）基于统计的方法。通过统计词频，位置等信息，计算句子权值，再简选取权值高的句子作为文摘，特点是简单易用，但对词句的使用大多仅停留在表面信息[8]。（2）基于图模型的方法。通过构建拓扑结构图，从而对句子进行排序。例如经典的 TextRank/LexRank[9]。（3）基于潜在语义的方法。使用主题模型，挖掘词句隐藏信息。如 LDA 算法[10]。（4）基于整数规划的方法。通过将文摘问题转为整数线性规划，求全局最优解[11]。目前，随着大数据、云计算等技术的发展，深度学习方法在 NLP 中取得了许多突破性的成果[12]。在文本摘要领域同样取得了很好的成就。SummaRuNNer 便是一个典型的文本筛选网络[13]，其将文本摘要重要句子提取问题，变为一个分类问题（二分类），提取的句子为 Label0，不提取的句子为 Label1。Wenpeng Yin 使用 CNN 进行文档筛选的建模，首先使用 CNN 建立一个无监督的 CNNLM 语言模型，通过该模型的训练可以将句子表示成一个稠密的向量，然后再进行文本筛选，得到目前最好的抽取式结果[14]。Jianpeng Cheng 通过 CNN 对句子进行压缩，变成稠密向量，然后将各个句子送入一个 LSTM 再利用基于 attention 的 LSTM 对每句话进行分类，该方法在长文本上取得较好的成果[15]。

目前，参与藏文摘要抽取的相关高校与研究人员较少，主要研究包括：安见才让通过提取文章五项特征对句子打分，从而形成摘要[16]，并提取基于敏感的藏文摘要[17]；南奎娘若根据藏文本身特点进行紧缩词还原等预处理步骤，在此基础上提取摘要[18]。本文的研究内容主要是面向单文档的藏文文本，通过构建图的方法为句子打分生成摘要，并且融合词向量挖掘深层语义，对生成的摘要使用 Rouge 方法进行评测。藏文自动摘要的研究对藏文信息化的发展，以及藏族文化的传播与交流都有着很大的促进作用。

1 研究方法

1.1 TextRank 算法

文本摘要可以利用文本信息本身的内容和结构特征实现,这类方法属于无监督的方法。除此之外，也可以通过大量的语料信息进行训练学习来抽取摘要，属于有监督的方法。这类方法不同于传统算法，实现简单，需要大量的训练数据。TextRank 算法是 PageRank 算法的一个变种，PageRank 是一种链接分析算法，Google 用其进行网页排序，是衡量网页重要程度的经典算法。Pagerank 算法大多用于有向图中，将指向该节点边的数量（前驱）与该节点指向其他节点的数量（后继）进行迭代。算法如公式 1 所示。

$$S(V_i) = (1-d) + d^* \sum_{V_j \epsilon In(V_i)} \frac{1}{|Out(V_j)|} \cdot S(V_j) \quad (1)$$

上式中 $In(V_i)$ 代表第 i 个节点的入度，即连接到该网页的网址数量，$Out(V_j)$ 表示第 j 个节点指向的网页数量。

将该思想应用于文本摘要的抽取中，通过衡量每一个句子与其他句子之间的联系，求出该句子的重要程度。对于一篇文档，传统算法大多忽略它的词语语义、语法等要素，简单地当成是词语的集合，并且每个词语都是独立出现的，互相不依赖彼此之间出现与否。如果将外部知识如语料库等信息融入到自动文本摘要的算法之中，能够改善摘要效果。具体做法如下。首先，TextRank 将一篇藏文新闻抽象成一个拓扑图 $G=(V,E)$，其中 V 表示图中所有节点集合，也就是新闻当中的句子，E 表示图中所有边，代表了句子之间的关系，利用 TextRank 对图模型进行迭代直至收敛，对所有节点进行排序，找出权重最大的句子作为文摘句。其过程主要分成四步：

1. 识别文本单元，并添加到图模型中形成节点。

2.识别文本单元间的关系，并添加到图模型中形成边。

3.对算法进行迭代直至收敛。

4.对节点进行排序，依据它们最后收敛时的得分。

迭代算法如公式 2 所示：

$$WS(V_i) = (1-d) + d^* \sum_{V_j \epsilon In(V_i)} \frac{W_{ij}}{\sum_{V_k \epsilon Out(V_j)} W_{ij}} \cdot WS(V_j) \quad (2)$$

其中 W_{ij} 表示节点 V_j 和节点 V_i 间边的权重，用节点 V_j 和节点 V_i 的相似度表示，我们采用不同的方法计算，$In(V_i)$ 表示指向节点 V_i 的所有节点集合，$Out(V_j)$ 表示节点 V_j 指向所有节点集合，d 为阻尼系数 $(0<d<1)$，一般取 0.85。一般在使用 TextRank 算法时，让所有节点初始得分为 1，并且某一节点误差小于 0.0001 的时候停止迭代。

在本文中引入了一些方法对其进行优化，在新闻文章中靠前的句子，以及与标题相似度较大的句子，摘要的可能性较大，所以适当增加权值。对于生成的藏文摘要 $S=\{s_1,s_2,...,s_n\}$,权值高的

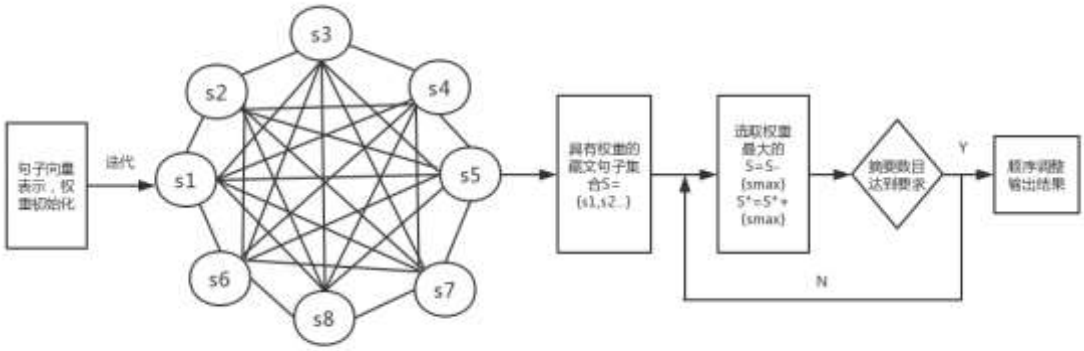

图 1 TextRank 算法流程

几个句子相似度一般很大，为了避免生成句子摘要冗余性很大的问题，我们引入惩罚系数，具体过程如图 1 所示。

1.2 词向量

自然语言处理（NLP）中最直观，也是目前为止最常用的词表示方法是独热表示方法（One-hot Representation），这种方法把每个词表示为一个很长的向量。这个向量的维度是词表大小，其中绝大多数元素为 0，只有一个维度的值为 1，这个维度就代表了当前的词。这种传统的方法有两种缺点，第一，向量的维度会随着句子的词的数量类型增大而增大，会造成数据稀疏以及维数灾难等问题。第二，任意两个词之间都是孤立的，仅仅将词符号化，根本无法表示出在语义层面上词语词之间的相关信息。

1954 年，Harris 等人提出的分布假说为词的分布式表示提供了理论基础：上下文相似的词，其语义也相似。Firth 等人在 1957 年对分布假说进行了进一步阐述和明确：词的语义由其上下文决定（a word is characterized by thecompany it keeps）。所以词的分布式表示可以更好地刻画原文的语义信息，例如国王(རྒྱལ་མོ)-男人(སྐྱེས་པ),在词的分布式表示的向量空间中与皇后(བཙུན་མོ)-女人(ཁྱིམ་མོ)的向量相近(见图 2)。

词的分布式表示根据建模主要分三类，其中基于神经网络的表现效果最好，基于神经网络的分布式表示称为词嵌入(Word embedding)，神经网络词向量表示技术通过神经网络技术对上下文，以及上下文与目标词之间的关系进行建模。在词向量中包含丰富的语义信息。将词向量与 TextRank 算法相结合，可以从语义层面上抽取文

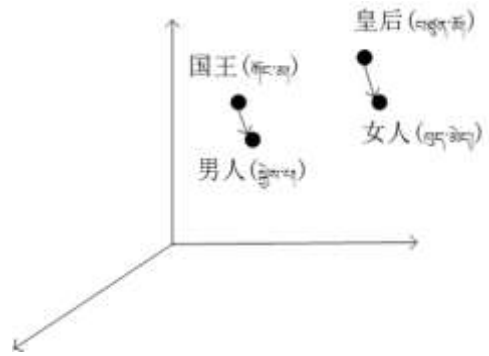

图 2 词的向量空间距离

章信息。本文中使用两种词向量模型，Word2vec 与 FastText。前者于 2013 年由谷歌开源，利用语言模型将词转化为向量表示，其过程如下

第一步:将 one-hot 形式的词向量输入到单层神经网络中，其中输入层的神经元结点个数应该和 one-hot 形式的词向量维数相对应。例如我们词典维度为 1000，则输入层包含 1000 个神经元。

第二步：通过神经网络中的映射层中的激活函数，计算目标单词与其他词汇的关联概率，其中在计算时，使用了负采样（negative sampling）的方式来提高其训练速度和正确率

第三步：通过使用随机梯度下降（SGD）的优化算法计算损失

第四步：通过反向传播算法将神经元的各个权重和偏置进行更新

FastText 是 Facebook 在 2016 年开源的分类器，其特点是速度快，FastText 方法包含三部分：模型架构、层次 Softmax 和 N-gram 特征。FastText 模型输入是一个词的序列（一段文本或者一句话），输出这个词序列属于不同类别的概率。

序列中的词和词组组成特征向量，特征向量通过线性变换映射到中间层，中间层再映射到标签。FastText 在预测标签时使用了非线性激活函数，但在中间层不使用非线性激活函数。

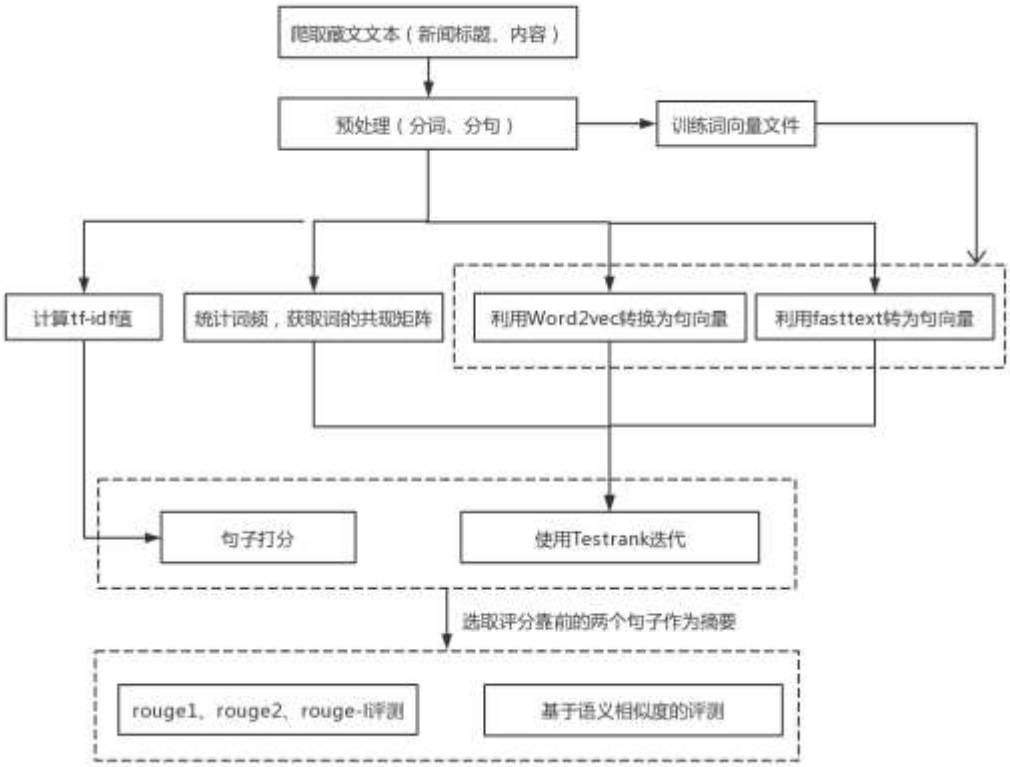

图 3 系统流程

1.3 评测方法

摘要评测从广义角度上可以分为内部评价法以及外部评价法[15]，目前文本摘要评测方法主要是由 Chin-Yew Lin 等人参考了机器翻译的自动评价方法 BLEU[19]，提出的 ROUGE(Recall-Oriented Understudy for Gisting Evaluation)评价方法[20]。该方法首先由多个专家分别生成人工文摘，构成标准文摘集。然后将系统生成的自动文摘与人工生成的标准文摘相对比，统计二者之间重叠的基本单元的数目，来评价文摘的质量，该方法现已成为文摘评价技术的通用标准之一。ROUGE 系列评价指标包括 ROUGE-N、ROUGE-L、ROUGE-S、ROUGE-W。其中 ROUGE-N 是最为常用的指标，ROUNGE-N 是基于 n-gram 共现统计，n 一般为[1,4]，计算公式如 3 所示。

$$ROUGR-N=\frac{\sum_{S\epsilon\{\mathrm{Ref\ summaries}\}}\sum_{n-gramsS}Count_{match}(n-gram)}{\sum_{S\epsilon\{\mathrm{Ref\ summaries}\}}\sum_{n-gramS}Count(n-gram)} \quad (3)$$

其中 $\mathrm{Ref\ summaries}$ 代表参考摘要，$Count(n-gram)$ 表示参考摘要中 $n-gram$ 的个数，$Count_{match}(n-gram)$ 代表生成的摘要中，与参考摘要中共同包含 $n-gram$ 的个数。而 $ROUGE-L$ 是基于最长公共子串的统计，$ROUGE-S$ 是基于顺序词对的统计，$ROUGE-W$ 是在 $ROUGE-L$ 基础上，考虑了串的连续匹配。不同的方法，在不同类型的文摘评测上有着不同的效果[21]。

2 模型构建

2.1 系统架构

基于 TextRank 的自动文本摘要算法的思想是将文本摘要的提取过程转换成文本中句子重要程度的排序过程。首先根据词向量模型训练语料库得到词语的词向量转换获得句向量，然后根据句向量计算句子之间的相似度，构建候选句子网络的图模型，即完整的句子之间的概率转移矩阵，通过迭代运算获取节点的重要性，实现自动文摘的排序和抽取。系统构建流程如图 3 所示。为了更直观的对比文本摘要质量，我们选取每篇文章首段第一个句子与最后一个句子作为一个简单的基准系统作为对比试验。

2.2 藏文文本预处理

结合藏文本身的特点，我们对爬取的语料进

行预处理，首先我们采用 CRF 的方法对藏文进行分词，并过滤掉停用词，建立词表，并按照藏文边陲符号”།”进行分句，对新闻中出现中文乱码，或者只有一到两个句子的噪声新闻用查找$utf8$的方式进行过滤。清洗之后的数据我们来训练词向量，这里我们采用两种不同的词向量模型，Word2vec 以及 FastText，将句子映射到高维词库中表示成向量形式。并根据词向量文件将每一条句子转化为句向量。

在 TF-IDF 方法中，我们直接通过将句子中每一个词的 TF-IDF 值相加，作为该句权重。TF-IDF（term frequency–inverse document frequency）指词频-逆文档频率，用于评估词对于一个文件集或一个语料库中的其中一份文件的重要程度。我们将句子中每一个单词的 TF-IDF 值相加，可以获得句子在新闻中的重要程度。TF-IDF 计算公式如(4)、(5)、(6)所示。

$$TF = \frac{n_{i,j}}{\sum_k n_{k,j}} \qquad (4)$$

$$IDF = \log \frac{|D|}{D_i + 1} \qquad (5)$$

$$TF - IDF = TF \times IDF \qquad (6)$$

$n_{i,j}$代表抽取句子中包含词的数量，$\sum_k n_{k,j}$表示新闻中包含该词总数量，$|D|$代表需要抽取的新闻总数目，D_i代表需要抽取的句子中包含该词新闻数。由于较长的句子包含的信息较多，所以 TF-IDF 方法抽取的句子长度一般很长，本文中只选取权最高的两个句子，并不考虑句子长度的影响。

2 3 句子相似度计算

基于 TextRank 的摘要抽取需要将篇章构造为无向图，节点表示句子，边表示句子之间的相似度，经过预处理后已经将句子转化为词语的集合，计算句子S_i和S_j相似度时，采用余弦距离表示（S_i和S_j都是使用高维词库映射的向量），如公式(7)所示。

$$WS(S_i, S_j) = similarity(S_i, S_j) = \cos(S_i, S_j) \quad (7)$$

这里我们也采用词的共现程度方式计算相似度进行对比，这种方法不需要将句子表示为高维的向量，分词之后根据两个句子中共同出现词语个数的平均值作为边的权重，从而构造出词的共现矩阵。

2 4 基于语义的结果分析

针对传统摘要方法不能很好地展示出系统摘要与参考摘要语义层面的相似度，使用词向量平均的方式来获取摘要语义层面的效果，具体计算方式如公式(8)所示。

$$sim(Ref, Sys) = \cos\left(\frac{\left(\sum_{i \epsilon Ref} vec(i)\right)}{len(Ref)}, \frac{\left(\sum_{i \epsilon Sys} vec(i)\right)}{len(Sys)}\right) \quad (8)$$

其中，Ref代表参考摘要，Sys代表系统摘要，$vec(i)$代表词i的高维向量表示，$len(\mathrm{Re}f)$代表摘要中词的个数，在这当中需要过滤掉摘要中未出现在词向量中的词。通过求两个平均向量的相似度，我们就可以获得摘要语义层面效果。

3 实验设计与结果分析

3.1 实验数据准备

目前文本摘要的数据集主要有两种，第一种是人工撰写的 DUC 数据集数量较少，第二种是 Gagiword 数据集,使用标题作为参考摘要，由于藏文缺乏摘要人工撰写数据集，本文采用第二种方式，使用新闻作为语料，标题作为参考摘要。新闻采用中央民族大学自然语言处理实验室在民汉舆情汇聚与分析项目中从人民网、新华网等网站爬取的 5000 篇藏文新闻作为语料,使用新闻标题作为参考摘要。对于每一篇新闻使用藏文分词开源工具 tip-las[21]进行分词之后，每一个单词在预先建立好的词表中进行查找，若不在词表当中则去掉该词。对于 TF-IDF 以及词频方法，我们采用 NLTK 包中的数据处理方法统计出文档中每一个词的词频以及 TF-IDF 值，对每一篇藏文新闻中句子进行权值计算。

我们实验中采用两个词向量模型，Word2vec 与 FastText，前者使用基于 python 语言的自然语言处理库 Gensim 中 Word2vec 模块，采用 CBOW 模型、维度为 100、窗口大小为 5 等默认参数对该文本数据集进行学习训练得到 165M 词向量模型文件。后者采用 FastText 官方提供的由藏文维基百科训练出的词向量，大小为 156M。维度为 300 维。

3.2 评测方法

我们选取新闻中的两句评分最高的句子作为

系统摘要，采用 pyrouge 包与 ROUGE-1.5.5 工具，以传统的 Rouge-1、Rouge-2、Rouge-l 方法中的准确率，召回率以及 F1 值对系统摘要进行评价，rouge 的准确率 p 表示系统摘要与参考摘要的共现词汇在参考摘要中出现的比例，r 代表系统摘要与参考摘要的共现词在系统摘要中出现的比例。并通过本文提出的基于句向量相似度的方法从语义层面上评估摘要效果。

3.3 实验结果分析

我们从新闻中抽取两个分值最高的句子作为摘要进行评测，Rouge-1、Rouge-2、Rouge-l 评测结果如表 1 所示。

从中可以看出，相比于 FastText，Word2vec 的效果更好一些，这也有一部分来自于官方 FastText 的向量掺杂许多英文与中文的无关词向量导致。基于 TF-IDF 的方法与基于词频的方法效果类似，并且其摘要也大致相同。由于我们只考虑句子权值，对句子长度没有要求，所以使用 TF-IDF 方法抽取的句子长度都很长，评分也较高一些。

基于句子相似度评测结果如图 4 所示，由于 FastText 维度较大，所以结果中与 Word2vec 的方法值相差较大，但是在一定程度上也反映了摘要的预测结果。

表 1 Rouge 评测结果(单位:%)

	Rouge1			Rouge-2			Rouge-l		
	P	R	F	P	R	F	P	R	F
TF-IDF	11.2	18.7	14.9	7.9	4.7	12.6	12.5	16.2	14.3
TR+Tf	13.5	21.3	17.4	8.0	10.4	9.2	15.9	21.6	18.7
TR+Word2vec	18.8	32.7	25.8	9.6	18.9	14.2	16.6	29	22.8
TR+ FastText	15.0	26.6	20.8	5.7	11.1	8.4	12.8	22.6	17.8
benchmark	13.9	14.2	14.0	5.0	7.3	6.2	13.0	13.2	13.1

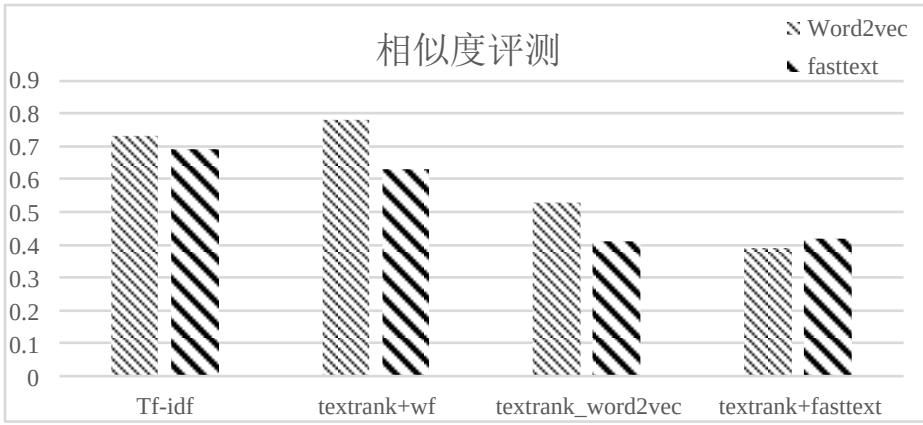

图 4 相似度评测结果

抽取摘要例子如下：

原文	ཤེན་ཀུ་ དབྱལ་བ་ ཞི་ རྩོང (ཚུས)19 རེ་ དལ་མཚོན་ ཐྱི་ དཀའ་གནད་ ནང་ གི་ དཀའ་གནད་ དང་ ། དཀའ་དལ་ ནང་ གི་ དཀའ་དལ ། འགག་སྐྱེ་ ཞི་ ནང་ གི་ འགག་སྐྱེ་ བཙས་ ཡིན་ པ་ མ་ཟད ། མཉེགས་ཤ་ ར་ གཏུགས་ དཀའ་ཤོས་ ཤེག་ ཀྱང་ རེད ། རིམ་པ་ ཁག་ དང་ ཐེ་ཚན་ ཁག་ གེས་ ཡིད་ཆེས་ དང་ ཚོར་ལེམས་ བརྟན་ དུ་ བཏང་ སྟེ་ ཡོའུ་ལན་ མཁར་རྩོང་ མ་ ཟླངས་ བར་ དུ་ ཐྱིར་མ་ཡོག་ པ་ ཞི་ དཔའ་ཤ་མས་ དང་ ‘ གཞན་ ཀྱེས་ སྐོབས་ཤུགས་ ཏིལ་ཙམ་ བཏོན་ ཚེ་ རང་ གེས་ སྐོབས་ཤུགས་ རི་ཙམ་ བཏོན་ པའི ’ ལས་ཀ ཞི་ ཏུར་ལེམས་ དང་ ।…… ནུས་ཤུགས་ ཡོད་ཆུ་ ས་ འགག་སྐྱེལ་ དང་ རྒྱལ་ཁ་ ཐུ་ ཐག་གཅོད་ པ ཞི་ དམག་འཐབ་ ཀྱི་ རྣམ་པ་ ཞེག་ ཚགས་པ་བྱས་ ནས་ ས་སྐྱ་ རྩོང་ གི་ དབྱལ་སྐྱེལ་ འགག་སྐྱེལ་ ལ་ རྒྱལ་ཁབ་ ཐུ་ ཐག་གཅོད་ ཚེང་ རྣང་གཞི་ བརྟན་པོ་ ཞེག་ བཏིང་ ཡོད ། གོམ་སྐྱབས་ རྩེས་མ ཞི་ ལས་དོན་ ལ་ དམིགས་ ནས་ ལེའུ་ཁྲེན་རྩེའུ་ ཡིས་ གཅིག་ ནས་ འགན་འཁྲི་ དོན་འཛུལ་ ཤུགས་ཚེན་ ཁྱེད་ དགོས་ ཤེང ། ཚབ་སྟིད་ ཐྱི་ ཚུགས་གནས་ ཐུར་བས་ མཐོ་ རུ་ གཏོང་ བ་ དང་ ། དབྱལ་སྐྱེལ་ ཐྱི་ སྐུས་ཚད་ མཐོ་ རུ་ གཏོང་ བ ། དལ་དུབ་ དང་ དམག་འཐབ་ ལ་ སྤུན་སྐྲང་ སྟེ་ བ ཞི་ བསམ་སྟོ་ ཁྱད་བསད་ཁྱེད་ ཐ ། རྩོང་ དང་ ཤང་རིམ་ གཉེས་ ཏང་སྟིད་ ཐྱི་ འགོ་ཁྲིད་པ་ ཚང་དང་ཡོ ཞི་ ཁྱེད་ཡོ ཞི་ འགན་འཁྲི་ ནན་ ཏུ་ གཏོང་ བ ། སྐུ་ངོ་མ ས་ ཏུས་འགོད་ དང་ ।....
摘要	Sen1:མཉེགས་ས ར་ གཏུགས་ དཀའ་ཤོས་ ཤེག་ ཀྱང་ རེད Sen2:ངལ་དུབ་ དང་ དམག་འཐབ་ ལ་ སྤུན་སྐྲང་ སྟེ་ བ ཞི་ བསམ་སྟོ་ ཁྱད་བསད་ཁྱེད་ བ
标题	རྒྱལ་ཁ་ ཐུ་ ཐག་གཅོད་ པ ཞི་ ཁྱེད་ཐབས་ ལ་ བརྟེན་ ནས་ མཉེགས་ས ར་ གཏུགས་ཤོད་པ་ཁྱེད་ བ །

4 总结与展望

本文提出了一种基于词向量加权的藏文自动文本摘要算法，利用 TextRank 算法将文档转化为图模型，使用不同的词向量模型生成句向量，利用句向量之间的相似度进行迭代，抽取摘要，首次使用 Rouge 方法对藏文摘要进行评测，为藏文抽取式摘要提供了基准。并在此基础上提出了基于词向量平均相似度的评测方法从语义层面上评估系统生成摘要的效果。由于 TextRank 算法属于无监督算法，我们在实验中没有使用较大规模的预料，有一定的局限性。下一步工作中，我们将采用深度学习的方法去抽取摘要，并尝试生成理解式摘要，将句子向量相似度的评测方法应用于理解式摘要当中，使摘要的评测更加规范。

参考文献

[1] Mani I. Advances in Automatic Text Summarization[M]. MIT press, 1999.
[2] 宗成庆.统计自然语言处理[M].清华大学出版社,2013.
[3] 洪冬梅. 基于 LSTM 的自动文本摘要技术研究[D].华南理工大学,2018.
[4] Cheng J, Lapata M. Neural Summarization by Extracting Sentences and Words[J].2016.
[5] Radev D R, Allison T, Blairgoldensohn S, et al. MEAD - A Platform for Multidocument Multilingual Text Summarization[C]// 2004.
[6] Woodsend K, Lapata M. Automatic generation of story highlights[C]// Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010.
[7] Luhn H P. The Automatic Creation of Literature Abstracts[J]. IBM Journal of Research and Development, 1958: 159-165.
[8] Brandow R, Mitze K, Rau L F. Automatic condensation of electronic publications by sentence selection[J]. Information Processing & Management, 1995, 31(5): 675 685.
[9] Mihalcea R, Tarau P. TextRank: Bringing order into text[C]//Proceedings of the 2004 conference on empirical methods in natural language processing. 2004:404-411.
[10] 孙国超. 基于 LDA 主题模型的 web 文本自动文摘系统的研究与实现[D].山东科技大学,2017.
[11] 解艳. 基于 LSA 和段落聚类的自动文摘系统的研究[D].辽宁科技大学,2012.
[12] Robert Dale. NLP commercialisation in the last 25 years[J]. Natural Language Engineering,2019,25(3).
[13] Nallapati R, Zhai F, Zhou B. SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents[J]. 2016.
[14] Yin W, Pei Y. Optimizing Sentence Modeling and Selection for Document Summarization[C]// International Conference on Artificial Intelligence. AAAI Press, 2015.
[15] Cheng J, LapataM. Neural Summarization by Extracting Sentences and Words[J].2016.
[16] 安见才让.藏文搜索引擎系统中网页自动摘要的研究[J].微处理机,2010,31(05):77-80.
[17] 南奎娘若,安见才让.基于敏感信息的藏文文本摘要提取的研究[J].网络安全技术与应用,2016(04):58-59.
[18] 南奎娘若. 基于特征信息提取的藏文自动文摘研究[D]. 2016.
[19] Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002: 311 318.
[20] Lin C Y. Rouge: A package for automatic evaluation of summaries[C]// Proceedings of the Workshop on Text Summarization Branches Out, 2004.
[21] 李博涵,刘汇丹,龙从军,等.基于深度学习的藏文分词方法[J].计算机工程与设计,2018,39(01):194-198.

李维(1995-)，硕士研究生，主要研究领域为文本摘要。
E-mail：1289773612@qq.com

闫晓东（1973-），通讯作者，博士，副教授，主要研究领域为少数民族语言信息化处理、自然语言处理。
E-mail：yanxd3224@sina.com

解晓庆（1996-），硕士研究生，主要研究领域为自然语言处理
E-mail：xiexiaoqing28@126.com