硕士专业学位论文

基于句法表征的专利文本相似性评估

Patent Text Similarity Assessment Based on Syntactic Representation

作者：陈泽龙

导师：郑宏云

北京交通大学

2019 年 6 月

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：陈泽秋　　导师签名：郑宏云

签字日期：2019 年 6 月 2 日　　签字日期：2019 年 6 月 3 日

学校代码：10004 密级：公开

北京交通大学

硕士专业学位论文

基于句法表征的专利文本相似性评估

Patent Text Similarity Assessment Based on Syntactic Representation

作者姓名：陈泽龙 学 号：17125009

导师姓名：郑宏云 职 称：副教授

工程硕士专业领域：电子与通信工程 学位级别：硕士

北京交通大学

2019 年 6 月

致谢

本论文的研究工作是在导师郑宏云副教授的悉心指导下完成的。郑宏云老师严谨的态度、开阔的视野以及渊博的知识给予了我极大的帮助。郑宏云老师谨重严毅的学术精神、恪尽职守的工作作风，深深地感染和激励着我不断前进，对我品行的塑造带来了很大的影响。在此衷心感谢两年来郑宏云副教授对我的悉心指导和关怀。

感谢实验室里的所有老师。衷心感谢赵永祥老师、李纯喜老师、郭宇春老师和陈一帅老师在我研究生学习阶段给予我的无私帮助和关怀，我所有的研究成果都凝结着各位老师的汗水。在此向各位老师表示诚挚的谢意。

另外，在实验室工作和撰写论文期间，张大富师兄、贾海涛师兄、曾显珣、杨晶晶、刘一健、苏健、高志朋、冯梦菲、赵红娜、刘子可师弟、李小乐师妹等同学对我的研究工作给予了热心的帮助，在此向他们表示真心的感谢。

另外，非常感谢国家自然科学基金项目《基于熵理论的信息匹配网络测量与建模》（基金号：61872031）的支持，它使我的研究工作有了更多的思路。

最后，特别感谢一直无微不至的关心、支持我的父母和女朋友，正是他们热情的鼓励和默默的奉献，使我得以顺利完成学业。

摘要

当下，对于专利相似性的研究非常重要。当用户申请新专利时，他们需要在专利数据库中进行相似专利检索，以进行专利查新，并且防止专利侵权，还可以从相似专利中获得灵感。因此这就对专利相似性评估提出了一定的要求。

专利主权项是专利文本的核心内容，全面阐述了本专利所保护的技术范围，专利相似的判定一般以权利主权项为标准。本论文对专利主权项文本进行深入研究，基于专利文本的 SAO（Subject-Action-Object, 主谓宾）句式特点，提出了一种基于句法表征的专利文本相似度算法。本论文基于这样一种假设：相似专利之间会出现相似关键词和相似句子。通过文本挖掘技术，挖掘出专利文本中的关键词来表征文本的含义。首先，通过关键词语义信息和句子结构特征计算专利文本之间的句子相似度，然后通过专利文本之间的句子相似度计算专利文本相似度。

本文的主要工作如下：

首先，利用文本挖掘技术提取专利文本中的关键词，对于分词效果不佳的关键词，总结其构词规律，利用基于规则的命名实体识别技术进行提取。

然后，考虑到专利文本包含大量 SAO 或 SA（Subject-Action, 主谓）或 AO（Action-Object, 动宾）结构，将文本切割成具有上述结构的“子句”集合，结合“子句”中关键词的语义信息和关键词的位置信息利用稳定匹配算法计算专利文本之间各个“子句”相似度。

最后，由于文本中的“子句”具有序列性，文本中的前后“子句”存在联系，所以将专利文本“子句”集合视为时间序列集合，利用 DTW（Dynamic Time Warping，动态时间归整）算法通过比较“子句”序列之间的相似性计算专利文本相似性。

最终，通过实验验证了本算法的有效性。实验结果表明本文提出的这种针对专利文本句式结构所制定的专利文本相似度算法相对于传统算法效果更好。

关键词：专利相似度；语义信息；句式结构；稳定匹配算法；时间序列；DTW 算法

ABSTRACT

At present, the research on patent similarity is very important. When the user applies for a patent, they need to conduct similar patent searches in the patent database to prevent patent infringement, to conduct patent novelty search and to get inspiration from similar patents. Therefore, this puts forward certain requirements for the retrieval of similar patents.

The claim of the patent text is its core content, which comprehensively expounds the technical scope protected by the patent. Patent similarity is usually judged based on the claim of the patent. This paper conducts an in-depth study of the claim of the patent. Based on the SAO sentence features of patent texts, a patent text similarity algorithm of syntactic representation is proposed. This paper is based on the assumption that similar keywords and sentences will appear in similar patents. Firstly, the sentence similarity between patent texts is calculated by the keyword semantic information and structure characteristics of the sentence, and then the patent texts similarity is calculated by the sentence similarity.

The main work of this paper is as follows:

Firstly, the text mining technology is used to extract the keywords in the patent texts; for the words with poor effect of word-segmentation, after summarizing its word-formation rules, a rule-based named entity recognition technology is proposed to identify them accurately.

Then, considering that the expression of the sentence in the patent text has a syntax structure of Subject-Action-Object (SAO), Subject-Action (SA) or Action-Object (AO), this paper proposes a kind of Chinese patent text similarity algorithm by cutting the patent text into a set of sentences. The semantic information of the keywords in the sentences and the position information of the keywords are comprehensively utilized to calculate the similarity between the patents' sentences.

In addition, since the "sub-sentence" in the text is sequential, there is a connection between the "sub-sentence" before and after the text. Therefore, the set of patent text sentences can be treated as a set of time series, and the similarity between patent texts is calculated by comparing the similarity between sentence sequences by the DTW algorithm.

Finally, we verify the effectiveness of this algorithm through experiments. The

experimental result shows that the patent texts similarity algorithm proposed in this paper is better than the traditional algorithm.

KEYWORDS: Patent similarity; Semantic information; Sentence structure; Stable matching algorithm; Time series; DTW algorithm

目录

缩略词表

英文缩写	英文全称	中文全称
LSTM	Long Short-Term Memory	长短期记忆网络
SVM	Support Vector Machine	支持向量机
TM	Text Mining	文本挖掘
VSM	Vector Space Model	空间向量模型
SAO	Subject-Action-Object	主谓宾结构
SA	Subject-Action	主谓结构
AO	Action-Object	动宾结构
LDA	Latent Dirichlet Allocation	隐狄利克雷分布
LSI	Latent Semantic Index	潜在语义模型
HMM	Hidden Markov Model	隐马尔可夫模型
CRF	Conditional Random Field	条件随机场
ME	Maximum Entropy	最大熵模型
TF-IDF	Term Frequency–Inverse Document frequency	词频-逆文本权重
EMD	Earth Mover’s Distance	搬土距离
WMD	Word Mover’s Distance	词移距离
DTW	Dynamic Time Warping	动态时间规整
SRMA	Syntactic Representation Matching Algorithms	句法表征匹配算法

1 引言

1.1 研究背景和意义

1.1.1 研究背景

在当下，专利至关重要。专利是技术进步和创新活动有用的知识来源[1]。据报道，企业的成功与其专利实力之间存在正相关关系[2-4]。事实上，新产品对公司发展的影响随着时间的推移而显著增加。20 世纪 70 年代，新产品占企业利润的 20%，但这个数字在 20 世纪 90 年代上升到 50%[5-6]。经济学家认为，新技术生产的产品为大多数国家国民财富带来了 40%至 90%的增长[7]。

当作者完成一篇新专利时，可以在专利库中查找相似专利，以进行专利查新，并且可以在通过学习相似专利，了解现有技术的发展，对自己的专利发明也能提供一定的思路。

同时，为了保护专利，需要防止专利侵权。随着专利数量的迅速增加，专利侵权案件也变得越来越频繁，专利侵权保护也变得越来越重要。为了防止专利侵权，首先就需要在专利局授予专利前，对提交的专利进行新颖性和创造性审查并检查是否已经存在相似专利，以此来判断该专利是否具有侵权行为。

目前的相似专利检索大多是基于人工完成的，由于专利库中专利数量巨大，如果仅仅通过人工查找的方式进行相似专利检索费时费力，并且由于对专业能力有一定的要求，查找准确率也难以达到一个较高的水准。

本论文基于以上背景，利用专利文本的语义信息和句式结构信息提出了一种基于句法表征的专利文本相似度算法，帮助找到相似专利。

1.1.2 研究意义

（1）现实意义

首先，查找相似专利可以为现有研究工作提供一定的思路，通过参考他人技术，为自己的发明创造提供灵感。

其次，专利能够为公司带来极大的经济效益和品牌效益，因此需要防止专利侵权。市场经济存在激烈的竞争，当企业对技术有了新解决方案后，希望在自己的产品迅速占领市场的同时，为其他企业使用本技术方案设置门槛。因此，专利

保护就变得格外重要了。申请专利的目的在于：第一，通过法律确定发明创造的归属，有效保护发明成果；第二，及时申请专利可以在瞬息万变的市场竞争中获得主动，防止竞争对手将同样的发明创造申请专利获得专利独占权，确保自身的产品生产和销售安全可靠。因此，需要保护已经发表相关专利的专利权人的权利，防止他人盗用他的专利侵犯专利权。

（2）理论意义

现有研究表明，专利文本中存在大量的 SAO（Subject-Action-Object，主谓宾）句式结构的句子[8]，然而经过后文的测量发现，专利文本中还包含大量的 SA（Subject-Action，主谓）结构和 AO（Action-Object，动宾）结构的句子。当前关于相似专利的研究主要基于专利文本的语义分析和文本结构分析。提取出专利文本中具有 SAO 结构的“子句”，通过比较专利文本之间的各个“子句”相似性来度量专利文本相似性。但是现有的研究又存在某些不足，例如：（1）无法准确的提取专利文本中的关键词；（2）没有充分考虑具有 AO 句式结构和 SA 句式结构的“子句”；（3）没有充分考虑专利文本 “子句”内部单词之间的关系以及单词对该“子句”的影响；（4）没有考虑组成专利文本的各个“子句”之间的关系。

本文将针对专利文本的语法结构特点进行讨论，在计算专利文本相似性时，充分利用专利文本的句式结构信息。

本文的主要贡献在于：

（1）为了更有效的提取专利的技术专有名词，相对于现有方法，本论文更加充分的考虑了专利文本中技术专有名词的构词特点，总结了更多专利文本中技术专有名词的构词规律，提出了一种基于规则的技术专有名词识别技术；

（2）充分考虑专利文本的句式表达，构成专利文本的“子句”不仅仅是 SAO 句式结构，还包括 AO 句式结构和 OA 句式结构。首先，“子句”内部关键词的顺序可以在一定程度上表示文本的上下文信息，并且“子句”中不同句法成分所对应的关键词对该“子句”的重要性程度不同。为了更加充分利用文本信息，本文根据专利文本“子句”中关键词之间的上下文关系和关键词对于“子句”的重要性程度对 “子句”中的关键词进行位置编码。并根据专利文本之间各个“子句”的关键词位置信息和语义信息设计出了更高效、更准确的专利“子句”相似度算法；

（3）众多“子句”共同构成了一篇专利文本，文本中的 “子句”具有前后文关系，所以文本可以看作 “子句”组成的时间序列集合。那么文本相似度问题可以转换为衡量两个“子句”序列相似性问题。通过计算时间序列相似度的方法计算专利文本“子句”序列相似度，能在更加充分利用专利文本的前后文信息的基础上得到专利文本之间的相似度。

此外，在专利文本中，权利要求书是说明要求本专利保护范围的专利申请文件。被批准的权利要求的内容限定了专利保护范围。判定他人是否侵权，也以权利要求为依据。权利要求书主要包括：专利主权项和专利从权项。由于，专利文本中的主权项简要描述了该专利所需要保护的所有技术范围，若两篇专利的主权项越相似，则这两篇专利所说明的技术越相近。同时，根据文献[9]，标题在很大程度上对文章的主旨进行了提炼。所以本文使用爬虫技术从知网上爬取了专利文本的标题和专利文本的主权项，通过比较专利文本之间标题相似度和主权项相似度的平均相似度来度量专利文本之间的相似度。

1.2 国内外研究现状

1.2.1 文本相似性的研究现状

当前国内外学者对于文本相似度的研究一般从以下三个角度进行。

第一，基于文本之间公共字符/字符串匹配的文本相似度算法，如果两篇文本共有的字符/字符串的数量越多则这两篇文本越相似。

第二，基于语义网络的文本相似度算法：利用词义网络中的层次体系结构或语义词典中的同义词来计算文本的相似度，如 WordNet[33]、HowNet[34]等。

第三，基于统计自然语言处理的方法：

（1）通过对文本进行特征学习：包括 LDA 主题模型（Latent Dirichlet Allocation，隐迪利克雷分布）[35]和 LSI 主题模型（Latent Semantic Indexing，潜在语义索引）[36]，通过将一篇文本映射到不同主题上，该文本对应不同主题的概率生成文本-主题向量，基于该向量计算文本语义相似度；

（2）基于语料统计的方法：通过计算不同概念在文中出现的频率以及概念之间的相关性来度量文本相似性[37-38]；

（3）基于深度学习的方法：将文本映射成一个向量，通过比较向量之间的距离计算文本相似度，如：Word2vec[39-40]、Doc2vec[41]。

许多学者已经对文本语义相似性度量作了大量的研究工作。金博等人提出了基于语义理解的文本相似度算法[42]，基于知网所提供的词义网络层次结构来度量单词之间的相似性，然后利用单词之间的加权相似度来计算段落之间的相似度，再然后通过段落之间的加权相似度计算文本相似度，与传统的文本相似性算法相比，准确性得到一定的提高。徐德智等人提出：通过衡量专利文本中的关键词在词义网络中的语义距离，来度量不同概念的语义相似性[43]。Chi 等人提出了一种本体模型文本相似度计算方法[44]，基于本体中关系上独特的概念模型特点，综合考

虑同义词词林和知网，提出一种混合词相似度算法，进而提出了一种基于本体模型的文本相似度算法。Aritsugi 等人提出了一种基于显性语义分析的文本语义相似度计算方法[45]，利用维基百科通过 Onehot 向量生成文本中出现的每一个单词的语义表示，则文本相似度可以通过文本中单词之间的相似度来度量。这些研究人员提出的文本相似度算法，在一定情境下都取得了可观的效果，可以说对于文本语义方面的研究是将来文本相似度研究的重要发展趋势。

本小节主要针对文本相似性的研究现状做了简要的叙述，后文将对专利文本相似性的研究现状做简要叙述。

1.2.2 专利文本相似性的研究现状

当前关于专利领域的文本研究主要包括：专利信息检索[10]、专利摘要[11-13]、专利技术趋势分析[14]以及专利文本自动分类应用[15]、专利文本相似性度量。本节主要针对专利文本相似性度量的研究进行简要阐述。

当前关于相似专利的匹配服务依赖于信息检索技术，通过分析专利文本的结构化数据（包括：作者、作者从属关系、专利技术领域、关键词等）来进行相似专利检索。然而该相似专利检索系统虽然易于理解且易于开发，但是不能理解专利文本的语义表达，因而不能充分利用专利文本的语义信息，导致所分析的文本信息的丰富性有限，在对专利的内容解释性方面受到一些限制。这是因为仅仅分析了专利结构化数据而忽略了包含更多专利细节的专利文本描述[16]。

此外，专利局将每个专利分类到它所属的技术领域后，通过人工阅读相关专利的方式在该技术领域中寻找相似专利。Grawe 等人使用 LSTM（Long Short-Term Memory，长短期记忆网络）对专利进行类别分类，相对于统计机器学习分类方法准确率有较大的提升[17]。尽管，仅仅在相同技术领域中寻找相似专利大大降低了人工成本，但是后续仍然需要以人工阅读的方式查找相似专利，费时费力。同时在当前技术发展的情况下，出现了越来越多的交叉学科，它们是对不同领域的技术进行融合形成的新技术。这导致对于那些具有交叉技术的专利而言，如果对其在技术领域上划分，某一个专利可能会分在多个技术领域下。而且不同的技术领域之间可能包含一定的技术重叠，因此技术上相似的专利可以具有不同的分类结果。因此，如果通过专利分类再查找相似专利，效率依旧不高。

除了上述的分类方法，还有一些人提出了基于相似专利的类比设计系统用来支持专利发明。类比设计是一种基于用户需求和专利之间进行相似性分析来推进新技术形成的方法[18-20]。根据用户的需求，推荐相关的专利，实现专利转化。因此，这就对需求和专利以及专利和专利之间的相似度计算提出了要求。早期的类

比设计系统主要是基于专利数据库开发的[21]。随着数据挖掘技术的发展，有人提出了基于构建专利地图的类比设计系统[22-23]。将专利集合可视化的映射到二维平面或三维空间，通过它们在空间中的距离关系来探索专利内容之间的联系。然而，他们使用的 VSM（Vector Space Model，空间向量模型）文本向量化方法不能考虑到文本的语义信息，同时在构建专利地图的过程中，对文本向量降维映射到二维平面无疑也会造成信息丢失。想要很好的计算需求和专利以及专利和专利之间的相似度，需要很好的提取出需求文本中说明具体需求的语句和专利文本中描述解决的问题的语句。但是，现阶段缺乏专利解决问题的识别系统。最近，由于文本挖掘技术的进步，有人提出了提取专利解决方案模式或术语关系的系统，但是同样存在上述问题。Tiwana 等人提出了一种识别专利所解决问题的搜索系统[24]，可以在一定程度上识别发明专利所解决的问题。作者认为问题解决概念常出现于“发明背景中”，作者为发明背景中的每个句子计算权重。权重计算规则基于以下两点：（1）基于专利文本的特点，在发明背景中越靠前和越靠后的句子越重要，因而权重越高；（2）作者设置自定义词典，自定义词典中包含那些重要的技术词汇，发明背景中的各个句子包含越多相关技术词汇，则该句子权重越高。在成功挖掘专利解决方案之后，比较需求和专利解决方案之间或专利解决方案之间的相似度，可以根据需求推荐相似专利以及计算专利之间的相似度。因此，这就对度量文本相似度计算提出了一定的要求。

随着自然语言处理技术的发展，计算文本相似度已经成为了可能。专利文献中的文本数据包含了大量的专利信息，对专利进行了较为详细的描述。通过分析比较专利文献中的文本数据可以较准确的度量专利相似度。同时由于 TM（Text Mining，文本挖掘）的发展，挖掘专利文本的关键词来表征专利文本信息，使得分析专利文献中的文本数据已成为可能。最近，TM 引起了越来越多的关注并且已经积极地应用于专利相似性分析。

Arts 等人通过提取专利文本关键词，利用 VSM 模型将专利文本向量化，然后基于 Jaccard 距离来度量两篇专利的相似性[25]。但是 VSM 模型无法表征出文本的语义信息，而且利用 Jaccard 计算文本相似度无法考虑到同义词的情况。这对于文本相似度的计算存在误差。

在文本挖掘和计算文本相似度时，越来越多的学者考虑到结合文本的语义信息，甚至还有学者考虑到了文本的语法结构。

在结合语义信息计算文本相似度时，最初人们都是利用 WordNet、HowNet、Freebase 等词汇语义网络。例如 WordNet，它是由自然语言处理工程师和语言学家共同构造的一种基于认知语言学的英语词典。它不仅仅将单词按照字母排序，而且根据不同单词的含义组建了一个超大型的“词义网络”，在“词义网络”中同

义词之间由于语义相近，所以在网络中的空间位置相近，同义词之间也会形成一个一个小型网络，每个同义词网络都代表一个基本的语义概念，网络之间通过各自的词义关系相连接。

Lee、Song 等人针对专利权利要求书的结构特点，提出一种对专利权利要求书构造树结构的方法[26]。该专利树的第一层是专利的标题；第二层是专利的各个权利说明，包括主权项、从权项；第三层是各个权利要求的实现方法描述；第四层是各个方法的细节描述。根据专利权利要求书不同部分之间的关系构建树结构中的不同层和不同节点，对专利树间同一层上的不同树节点中的权利描述一一进行相似度计算，最终加权得到专利相似度。但是，该方法仅仅只是粗略的计算专利权利要求书各个部分之间的相似度，没有应用到专利内部的语法结构，且作者在计算段落之间的相似度时利用的是 VSM 模型，没有考虑到语义信息。

Wang、Song 等提出了一种将文档表示为异质信息网络来计算专利文本相似度的方法[27]。首先通过文本挖掘提取出了文本中的实体词及其类别，然后基于 Freebase 数据库构建文本中实体词之间的关系，不同的文本通过它们实体词之间的关系相连接。因此，可以将文本相似性问题转换为文本之间元路径距离问题。相对于传统的异质信息网络仅仅只用文本的结构化数据，例如：作者、机构等，本论文充分利用了文本本身的信息。但是它极度依赖于 Freebase 世界知识库中存在的知识，倘若相关实体关系没有在 Freebase 世界知识库中出现，则会造成异质信息网络的信息丢失。

Sharma、Tripathi 等提出一种利用文本单词的语义信息计算专利文本相似度的方法[28]。作者根据单词在 WordNet 词汇网络之间的跳数，度量单词之间的相似度。通过文本中单词之间的相似度，最终加权平均得到专利相似度。该方法极度依赖 WordNet 词汇网络，然而 WordNet 无法穷尽所有可能会出现在文本中的单词，对那些新词，无法计算它与任何单词的相似度。同时该论文在通过单词相似加权求得专利相似时，未能考虑到专利文本内部的语法逻辑。

Wang、Cheung 等人结合文本挖掘技术和语义分析技术度量文本相似度，用于基于专利信息分析的知识管理系统[29]。作者认为专利文本中的技术专有名词包含重要的信息，作者通过挖掘专利文本中的关键词来表征专利文本信息，并利用 WordNet 中单词的上下位关系，对专利库中的所有专利所提取出来的关键词构建实体关系网络。若一个单词与越多实体词存在关系，则该单词对其所在的专利就越重要。通过实体词对于专利文本的重要性和专利文本之间存在关联的实体词数量来定量专利文本之间的相似度。但是，在构建实体关系时，仍然存在前述问题。

此外， Sharma、Tripathi 等人还提出一种考虑专利文本间各个单词在 WordNet 中的跳数距离，来定义实体之间的相似度[30]。当两篇专利中相似度超过某个阈值

的实体词达到一定的比例时，将它们视为相似专利。

张海超等人考虑到专利文本大多数是以 SAO 结构进行表述的，提出一种基于 SAO 的专利结构相似度计算方法[8]，提取专利文本中的 SAO“子句”，利用词义网络计算不同 SAO“子句”各个句法成分之间的语义相似度（“子句 1”的主语/宾语和“子句 2”的主语/宾语之间的相似度、“子句 1”的谓语和“子句 2”的谓语之间的相似度），为“子句 1”中的每个单词从“子句 2”中相应的句法成分中匹配到语义最相似的单词实现“子句”之间单词一对一的匹配，然而在进行一对一匹配时应该要综合考虑两个“子句”各个单词之间的相似度，为两个“子句”的各个单词寻找到一个全局最佳的匹配关系。同时，专利文本中不仅仅只有 SAO 句式，还存在一些诸如 SA 句式和 AO 句式的“子句”，如果仅仅只提取 SAO 句式进行相似度计算，忽略了专利文本中的大量信息。

在专利分析中应用文本挖掘最重要的优点是：专利文本中包含重要的研究成果，通过文本挖掘可以快速处理大量的专利文件并提取出文本的关键特征。文本挖掘已经广泛应用于协助专利工程师或决策者进行专利分析[24]。然而，专利文献对文本挖掘提出了独特的挑战。首先，大多数现有的文本挖掘算法不能区别关键词的同义词[31]；另外，文本挖掘算法难以识别复合词；此外，需要针对不同的文本，挖掘出较多的关键词，确保文本之间的细微区别[24]。对文本挖掘算法得到的关键词，由于外部词典仅包含有限的信息，因此基于构建词义网络来度量单词之间的相似性有一定的缺陷，故对单词之间的语义相似性也提出了一定的要求。

研究以上文献发现，当前关于专利领域的文本相似度分析大多是先通过挖掘专利文本中的关键词，利用关键词来表征专利文本的信息；然后再基于专利文本的关键词利用相关文本相似度方法度量专利文本相似性。以上文献都没有对专利领域的文本挖掘做什么优化，然而考虑到专利文本中的关键词大多数是合成词[32]，如果不针对专利文本的特定场景做一些优化，难以准确的提取出专利文本中的关键词；同时上述文献所使用的文本相似度算法都没有充分利用到专利文本信息，包括：（1）句式结构（2）专利文本中“子句”的“序列性”。

因此，基于当前的研究现状，本论文主要从两个方面着手分析文本相似度。第一，结合专利文本的表述方式，进行专利文本挖掘，提取专利文本关键词，提高专利文本挖掘的准确率；第二，从专利文本语义和文本结构角度进行专利文本的相似性度量。

1.3 论文组织结构

本论文共有六章，具体结构安排如下：

第一章为引言部分，主要介绍了本论文的选题背景以及研究意义，阐述本课题的研究现状，概述本论文的研究内容以及相关章节安排。

第二章为相关技术部分，首先介绍了文本挖掘的流程及方法，包括：分词、词性标注、命名实体识别、去停用词、关键词识别；其次介绍了 Word2vec 单词分布式表征技术；然后介绍了一些传统的文本相似度的研究方法，包括：VSM、Word2vec、Doc2vec、LDA、LSI、WMD（Word Mover's Distance，移词距离）；最后介绍了一种稳定匹配算法和 DTW（Dynamic Time Warping，动态时间规整）时间序列相似度算法。

第三章介绍了专利文本的特点，主要分为两个部分：（1）专利文本的构词特点，首先分析了专利文本中的技术专有名词的构词特点，并根据现有文本挖掘技术的不足，对于专利文本中的技术专有名词难以进行准确的分词的情况，总结出其构词模式，利用模式和字符串相匹配的方法提取出符合该模式的字符串作为合成型技术专有名词；（2）专利文本的语法特点，主要分析了专利文本中的基本句式结构的特点。为了更加充分的利用专利文本的语义信息，将专利文本切割成具有一定句法规律的“子句”集合，并结合现有针对专利文本语法结构的研究，对专利文本“子句”中的单词提出一种位置编码方式。

第四章介绍了本论文提出的专利文本相似性算法。本文基于 Word2vec 进行专利文本关键词之间的语义相似性计算。首先通过综合考虑专利文本“子句”之间各个单词的语义信息和位置信息度量专利文本各个“子句”之间的相似性，然后通过专利文本各个“子句”之间的相似性度量专利文本之间的相似性。

第五章为算法的实现和验证部分，通过对比实验将本文提出的算法与现有专利相似性算法进行比较，验证了本研究的有效性。

第六章为总结与展望部分，总结本论文所做的工作和当前研究的一些局限性，并对未来的研究方向提出了一些参考建议。

2 论文相关知识介绍

在各个研究领域，机器学习、深度学习、文本挖掘都得到了广泛的应用，其中就包括文本相似性度量。本章主要介绍了本论文的相关技术背景、主要方法以及目前的研究现状。

2.1 文本挖掘

我们身处一个“大数据”时代，生活中的方方面面都随时随地产生着海量的数据，它们中包含丰富的有价值的信息。这就对大数据分析与处理提出了一定的要求，我们称之为数据挖掘，数据挖掘主要过程如图 2-1 所示：

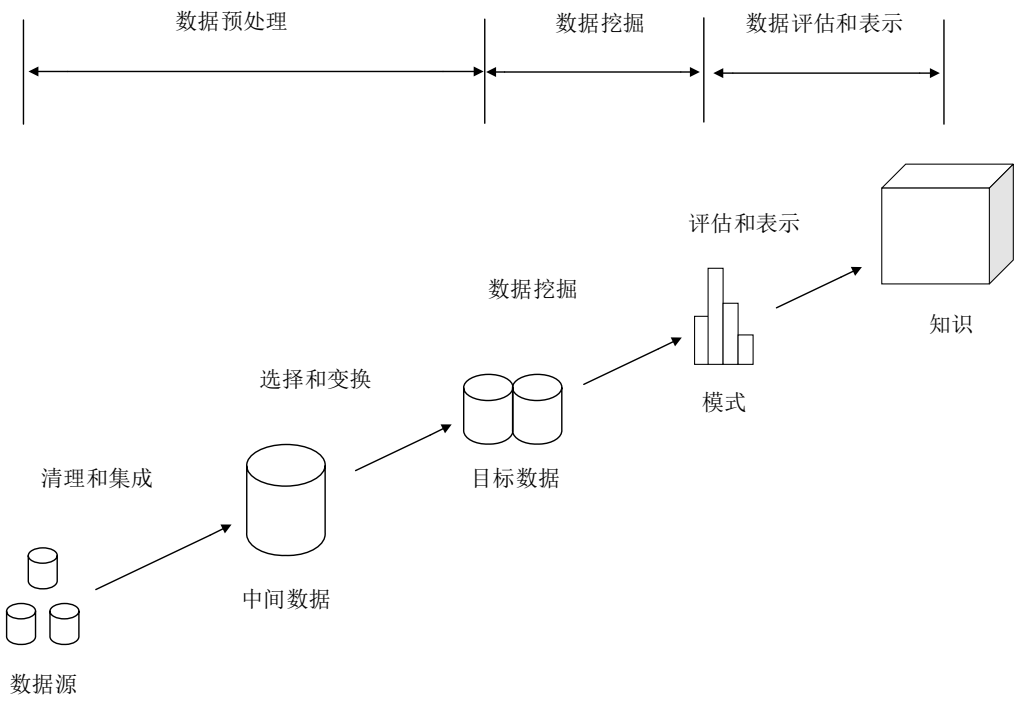

图 2-1 数据挖掘过程
Fig 2-1 The process of knowledge discovery in database

从大量的文本数据中获取有价值的信息和知识，我们称它为文本挖掘。文本挖掘的主要流程如图 2-2 所示，文本挖掘的主要操作步骤包括：（1）获取文本：从现有文本数据导入，或者通过网络爬虫等技术获取网络文本；（2）文本预处理：首先去除文本中的噪声数据用以优化挖掘精度，或者在文本数量极大的情况下，对文本数据进行重采样，仅选取其中一部分文本数据来提高挖掘效率；（3）然后

进行一些文本的语言学处理，包括：分词、词性标注、去除停用词、文本特征提取。文本特征提取的方法主要包括：1）用统计数学的方法，进行最具有代表性特征的提取；2）专家挑选；3）用非线性变换的方法将初始特征变换为有明显特点的新特征。

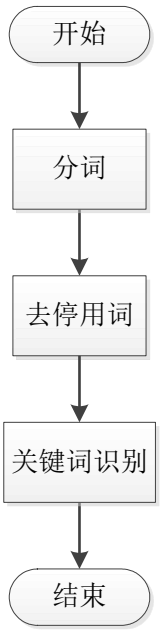

图 2-2 文本挖掘流程

Fig 2-2 The process of text mining in database

2.1.1 中文分词

中文文本分词是指将一段句子或文本等中文序列切分成一个个独立的词。它是对中文进行信息处理的基础，在自然语言处理领域具有重要的理论意义和应用价值[46]。现有分词算法主要包括：

（1）机械分词方法，该方法又叫字符匹配，它是通过字符串匹配来实现分词的算法[47]。首先需要人工建立一个分词词典。然后以某种程序扫描待分词文本，通过匹配文本中的字符串与分词词典中的单词，当某个字符串在分词词典中可以被找到时，则认为该字符串是一个独立的单词。按照扫描方式的不同，机械分词方法可以分为正向最大匹配法（方向从左至右）、逆向最大匹配法（方向从右至左）、最少切分算法（对每一句话中切分最少的单词）、双向最大匹配法（进行正向、逆向两次扫描）等。机械分词方法原理简单且相对容易实现，应用广泛。但是它极度依赖于分词词库，对那些未出现在分词词库中的单词无法进行准确的切分。但是将所有的单词加入分词词库是不现实的，而且机械分词方法没有从单

词语义上考虑，不能消除中文单词的歧义性。

（2）基于理解的分词方法，该算法考虑了中文单词的语义[48]。在分词的时候，通过对中文文本进行语义分析和句法分析，利用汉语中的语义信息和句式结构信息来实现单词的准确切分和语义消歧。通过让计算机模拟人对于句子的理解过程来正确识别单词。但是由于汉语逻辑十分复杂，目前基于理解的分词算法还处于预研阶段。

（3）基于统计的分词方法[49]。词是由汉字组合而成，并且形成词的方式是稳定的，因此在一段文本中，相邻的字共现次数越多，这些字越有可能构成一个词。当共现次数超过一定阈值时，便认为此字符串将有可能构成一个独立的单词。该分词方法可以在一定程度上解决单词语义消歧问题，并且能识别新词。但是，它会抽取出一些共现频率高、但是并不是词的常见字符串，例如“这一”、“我的”、“是一”等。同时因为没有添加分词词典，所以对常用词的敏感度较低。

（4）基于统计机器学习的分词方法。常用的算法模型包括：HMM（Hidden Markov Model，隐马尔可夫模型）、CRF（Conditional Random Fields，条件随机场）。通过人工标注大量已经分词的文本，利用机器学习算法训练标注文本，学习文本中的分词规律，从而实现对未知文本的分词。此方法的最大不足是需要大量预先完成单词切分的语料，且训练开销极大。当前，很多研究工作者在特定的语料下训练出了一套完整的分词模型，现有的开源工具包括：Boson 分词、Jieba 分词、Hanlp、哈工大 LTP 等。但是，如果直接使用开源工具包，一般在自己特定语料库下的分词效果不佳。

由于语言逻辑复杂，现有的分词方法还不能对一段文本进行准确的分词，但现有分词算法的准确率在某些语料上已经可以达到 95%以上。

2.1.2 词性标注

词性标注也称为语法标记或词类消歧，是指为分词结果中的每个单词根据与上下文之间的关系标注正确的词性[50]。

词性标注是一个让计算机理解文本组成的过程。目前常用的词性标注方法主要包括以下两种[51]：

（1）基于规则的词性标注方法，该方法基于语言学的研究成果，首先通过词性词典对语料进行切分然后对切分后得到的单词所有可能的词性进行标注，再依据单词的上下文环境，利用语言规则最终得出唯一适合的词性。但是由于该方法极度依赖于现有的语言学知识，相关规则需要人工构造，开发周期较长。且现有的关于语言学的研究有限，所构造的规则不能包含很多情况。

（2）基于统计学的词性标注方法，常见的词性标注算法包括 HMM、CRF 等[52-53]。基于统计的词性标注方法需要使用大规模语料库进行训练，好在已经有了很多开源工具可以使用。这种方法使用范围更广，标注结果一致性和覆盖率更高，现已广泛应用于各种文本处理任务中。

其实词性标注并不是文本预处理中的必需步骤，但是本文需要利用词性进行停用词去除与文本关键词的识别。

2.1.3 停用词去除

停用词是指在文本中出现频率较高，但是对文本的语义没有实质性影响的单词。在自然语言处理过程中，为了提高文本表征能力，可以根据任务需求去除文本中所包含的停用词。中文常见的停用词包括：“也”、“的”、“在”、“为”等等。

根据对文献[54-55]进行总结，现有的停用词去除方法包括基于停用词典和基于词性标注两类。

（1）基于停用词典的方法如下：根据文本处理任务的不同，停用词词典可以根据具体任务要求自行构建或使用现有的停用词典资源。将分词后的文本与停用词典进行一一匹配，若匹配成功则删除该单词。现在已经有很多研究机构公开了他们总结的停用词表，例如：“哈工大中文停用词表”、“百度停用词表”等其中包含了大量业内公认的停用词，可以直接下载使用，如果有不足，还可以自行添加。

（2）基于词性标注的方法主要借助文本词性标注算法，停用词的词性一般为连接词、标点符号、语气词、代词、介词五类单词。可以根据具体语料和文本处理任务的不同，对需要过滤的停用词的词性种类进行添加或删除。

2.1.4 命名实体识别

命名实体识别指识别出文本中具有一定含义的实体词，主要包括地名、人名、机构名、专有名词等。一般从两方面评价命名实体识别的效果：实体边界是否正确；实体类型是否正确。在专利语料中包含大量的技术专有名词，例如：多量子阱层、发光二极管、p 型氮化镓等。如果不做任何优化，直接对专利文本进行分词处理，这些技术专有名词通常难以准确切分。

命名实体识别存在诸多难点。英语中的实体词具有较为明显的标志：每个实体词的首字母大写，所以识别实体词的边界相对容易。

和英语相比，中文语料中实现命名实体识别主要存在以下难点：

（1）命名实体识别的第一步是进行正确的分词，然而中文文本没有同英文文本类似的空格之类的显式边界标志；

（2）对于一些从英语音译成汉语的专有名词，它们与常规的专有名词有着不一样的构词特征。

现有的命名实体识别的方法主要包括：

（1）基于规则的方法。基于规则的方法大多通过人工定义规则模式，模式特征包括关键字、指示词、方向词、统计信息、标点符号、位置词、中心词等方法。以模式和字符串相匹配自动提取出符合相应模式的字符串作为实体词，此方法依赖于预先建立的词典和知识库。当自定义规则能较为准确地反映语言现象时，基于规则的方法要好于基于统计的方法。然而这些自定义规则非常依赖于具体语言、领域和文本风格，构造规则一般周期非常长且不能包涵文本中所有的语言现象，极易发生错误，同时规则可移植性差，针对不同的文本集需要重新构造规则。另外，在通过设定一定的规则来进行命名实体识别时，不同的命名实体具有不同的构词特征，难以用一套规则来刻画文本中所有实体特征。

（2）基于统计的方法。主要包括 HMM、ME（Maxmium Entropy，最大熵）、SVM（Support Vector Machine，支持向量机）、CRF 等。对于这四种方法，最大熵结构紧凑，通用性强，主要缺点是收敛速度慢、训练时间长。条件随机场模型是一个全局最优、特征灵活的标注框架，但主要缺点在于收敛速度慢、训练时间长。同时也有人也提出了一种结合 CRF 和 LSTM 的命名实体识别技术，此方法收敛速度更慢、训练时间更长，但是由于利用到了 LSTM 神经网络，更能捕捉长文本内的前后依赖关系，通过利用更丰富的上下文信息，能够优化命名的实体识别。一般说来，ME 和 SVM 的正确率要比 HMM 高一些，但是由于通过 Viterbi 算法对于实体类别的识别效率更高，因此 HMM 更加适用于那些实时性高以及文本规模大的场景。

基于统计的方法对特征提取有较高要求，需要从文本中提取出实体词高质量的特征。需要对文本所包含的语言信息进行统计分析。相关特征包括：停用词特征、核心词特征、单词特征、词性特征等。同时，基于统计的方法需要大量标注语料，针对语料情况人工标注出文本中的各个实体的属性和边界，输入模型进行训练。但是，在当前条件下，没有现成的专利领域的大规模高质量标注语料库，想要通过人工标注的方式从专利文本中标注出高质量的技术专有名词，费时费力。

2.2 Word2vec 单词分布式表征技术

Word2vec 是由 Mikolov 等人提出的一种最新的词向量预训练模型，用来表征单词的向量表达[39-40]。它利用神经网络语言模型将每个单词转换为向量表示，在神经网络中经过大规模的语料训练所得到的词向量将具有良好的语义特性，具体表现在语义相近的单词的词向量在向量空间中距离较近，而语义无关的单词在向量空间中距离较远。模型的性能与训练语料的规模大小紧密相关，在一般情况下，语料越丰富最终的模型泛化性能越好。

Word2vec 模型本质上是一种无监督的浅层神经网络，以模型输入的不同，分为 Skip-gram 模型和 CBOW 模型，如图 2-3 所示。这两种模型都是三层神经网络结构，分别为输入层、映射层和输出层。与传统的神经概率语言模型相比，Skip-gram 模型和 CBOW 模型首次提出充分利用单词上下文信息的思想，在 Skip-gram 模型和 CBOW 模型中，不仅仅只考虑待预测单词之前的 k 个单词，而且还考虑待预测单词之后的 k 个单词。对每个单词进行上下文环境设置可以扩展每个单词的上下文信息，捕捉到单词更丰富的语义信息，提高单词向量化表征效果。以 Skip-gram 模型为例，每一个单词的训练目标都是最大化它在文本中与上下文单词的最大似然函数[39-40]：

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j\in nb(t)}\log p\left(w_j \mid w_t\right) \tag{2-1}$$

式中 $nb(t)$ 是当前单词 w_j 的上下文集合，称它为包含单词集合的滑动窗，共计有 T 个 ：$w_1 - w_T$ 。CBOW 与 Skip-gram 的区别在于：CBOW 模型是通过上下文来预测中心词，输入 $2k$ 个上下文单词对应的词向量，输出中心词所对应的词向量；而 Skip-gram 模型则与此相反，它是用中心词来预测上下文，输入中心词所对应的单词向量，输出 $2k$ 个上下文的单词向量。另外，CBOW 模型在输入过程中，对其输入的 $2k$ 个向量会在映射层进行求和，而 Skip-gram 模型一般对输入向量不做任何处理。

由于模型每输入一个滑动窗中的单词都要在输出层遍历词典进行 softmax 归一化，因此 Skip-gram 模型与 CBOW 模型计算成本非常高。当前，降低词向量特征表达的计算复杂度的优化方法主要包括：分层 softmax（Hierarchical Softmax）与负采样（Negative Sampling）。

分层 softmax 是针对输出层在词典 D 中计算 softmax 归一化需要耗费大量计算复杂度的一种高效的优化方法。它利用 Huffman 树结构表征词典中的所有单词，将词典中的每个单词映射到 Huffman 树中的叶节点上，如图 2-4 所示[56-57]。根据语料中单词出现的频率来调整该单词所在 Huffman 树中的层数，对于高频单词的叶节点所处的树层数较小，对于低频单词的叶节点所处的树层数较大。因此，每个

单词在 Huffman 树中都存在唯一一条从根节点到相应叶节点的路径，Huffman 树中的每一个内部节点都表示一个特征向量，因此这条路径表示输出单词对应的概率。采用分层 softmax 算法训练词向量时，在输出层仅仅需要更新对应路径上节点的特征向量，不需要遍历全局词典。因此，对于每个训练样本而言，分层 softmax 在输出层的计算复杂度从 $O(V)$降低到了 $O(\log(V))$ [39-40]。

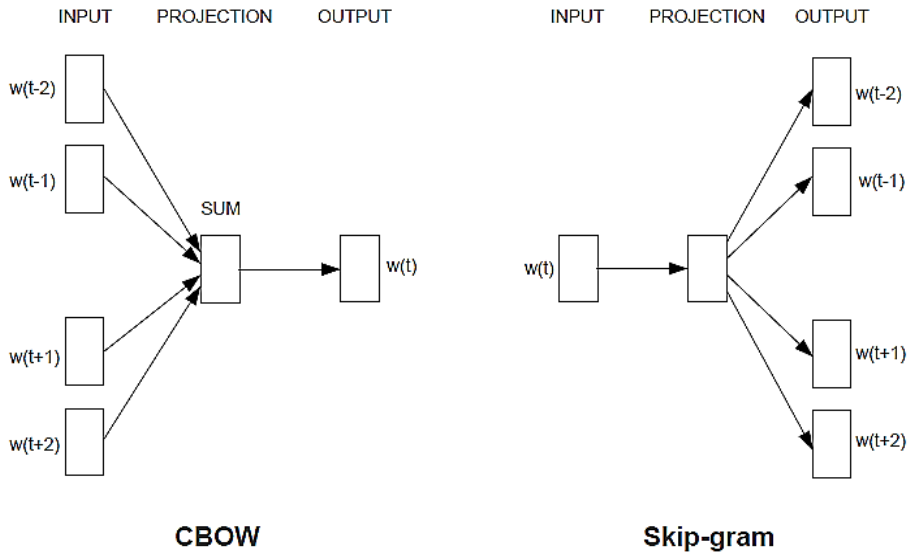

图 2-3 CBOW 与 Skip-gram 模型架构[39]

Fig 2-3 CBOW and Skip-gram model architecture[39]

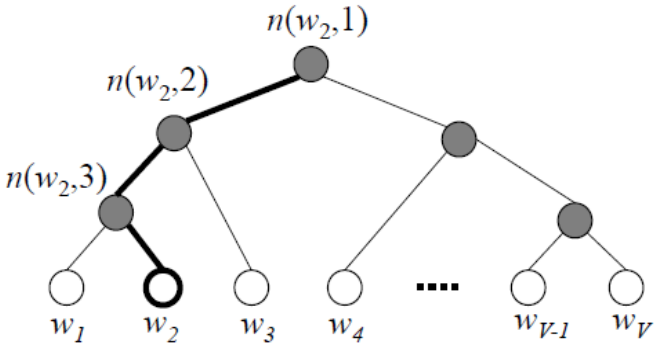

图 2-4 分层 softmax 结构[56-57]

Fig 2-4 The architecture of Hierarchical Softmax[56-57]

负采样算法是另外一种降低词向量训练计算复杂度的优化方法，现有的研究表明它的训练效果和优化效果都要优于 softmax。该算法将输出的目标单词视为正样本，将词典中其他单词视为负样本。为了降低输出层的计算复杂度，同时可以更好的区分不同单词，每当向模型输入一个目标单词，负采样算法都会以一定的概率从词典中抽取一定数量的负样本。与分层 softmax 采用的对数概率作为目标函数不同，负采样算法的目标函数如下式所示：

$$\log\sigma(v'^{T}_{w_O}v_{w_I})+\sum_{i=1}^{k}E_{w_i\sim P_n(w)}[\log\sigma(-v'^{T}_{w_i}v_{w_I})] \tag{2-2}$$

其中$\log\sigma(v'^{T}_{w_O}v_{w_I})$表示模型输出目标单词的概率，$\log\sigma(-v'^{T}_{w_i}v_{w_I})$表示模型输出非负样本单词的概率，$P_n(w)$表示负样本单词的分布。负采样算法在反向传播更新模型参数时只会利用与输入目标单词和 k 个负样本单词相关的参数，与 softmax 算法需要遍历全局词典来更新模型参数的方法相比，负采样算法可以极大的降低参数更新数量，从而极大的降低训练成本。

词向量一般具有固定的维数，如 50、100、200 维。Word2vec 词向量训练方法，完全基于输入语料来计算单词相似度。它以文本训练语料作为输入，以文本中各个词的词向量表征作为输出，将文本中每个单词量化为一个 N 维词向量。最终可以通过计算词向量之间的距离来计算各个单词之间的语义相似性。它的优点是能够快速准确的训练词向量。

本文使用 Word2vec 模型，将专利的主权项以关键词的形式作为输入，通过调整模型参数来达到优化词向量的训练效果。

2.3 文本相似度

在如今信息量飞速增长的时代，想要在一大批文本库中找到两篇相似的文档，如果通过人工阅读的方式完成检索，工作量大、效率低下。在当前阶段，已经有很多研究学者提出了一些文本相似度算法，本节将对一些主要的文本相似度算法做简要阐述：

（1）基于两篇文本共有字符的数量，若两篇文本共有的字符数越多，则这两篇文本越相似。然而，文本中通常存在很多同义词，这些同义词虽然表述不同，但是语义相似。在寻找相似专利时，存在大量的专利文本，由于作者的撰写方式不同，即使它们是相似专利，它们之间也有可能不存在任何相同的关键词，故不应仅仅比较两篇专利共同拥有的字符，而需要从语义层面进行分析。

（2）VSM 模型把对文本内容的相似度计算简化为向量空间中的向量的相似度计算[58]。它采用词袋模型和 TF-IDF（Term Frequency–Inverse Document frequency，词频-逆文本频率）将文本建模成词频向量，文本中存在多少个词，词袋模型所建模的向量就有多少维，每个词在高维向量中存在一维位置，因此文本中的各个词可以映射到高维向量中的各个位置上，各个位置上的值对应的是该词在文本中相对于文本集合的 TF-IDF 权值。将每个文本映射成一个高维向量后，计算向量之间的距离计算文本间相似度。很明显，利用 VSM 计算句子相似性的时候，并不是语义级别的计算，而是字面相似的计算，例如：假设两个句子分别出现“计算机”

和“电脑”，按照 VSM 模型它们是没有相似性得分的，但是它们之间的语义却是相似的。同时 VSM 模型得到的文本向量维度高，向量非常稀疏，最终向量之间容易正交化。可以看出其本质还是在比对字符。该算法对于实现语义级别的文本相似性分析的效果一般，对于专利主权项描述，它们之间文本结构不同、描述方式不同、且长短也不相同，这些都对实现语义级别的文本相似性分析提出了更高的要求。

（3）LDA 主题模型，该方法将文本集中每篇文本的主题按照概率分布的形式给出[35]。每篇文本对应各个主题都计算一个概率值，生成文本-主题向量。但是 LDA 非常依赖于先验知识，需要预先设定主题数 K，或者通过学习的方式让模型自动生成主题数 K，但是如果通过模型自动生成主题数的方式选定 K 值可能会导致过拟合。同时，通过计算不同文本之间文本-主题向量的距离来度量文本相似度，这样导致了计算出来的两个相似专利可能只是主题相似，在一定程度上专利的主题并不能反映专利的特性，我们希望能够搜索出技术特点相似的专利，这就对挖掘文本的语义相似性提出了较高的要求。

（4）基于 Word2vec 计算文本相似度，该方法通过深度学习模型训练得到文本中所有单词的词向量，对文本中所有词向量求和取平均，得到该文本的特征向量，进而通过计算文本向量之间的距离度量文本相似度[59]。虽然该方法从单词语义的角度度量文本相似性，但是将文本中的词向量求和取平均后，会使得文本中那些独一无二的关键词向量的特征与其他向量平均，特征信息被削弱。

（5）Doc2vec 原理与 Word2vec 相似，但是它是将文本以句子为单位，通过浅层神经网络将句子编码成句向量，通过计算句向量之间的余弦相似度或欧氏距离来计算文本相似度。相对于 Word2vec 将文本中各个单词向量取平均来表征文本向量不同的是，在生成句向量时模型考虑了单词在文本中出现的顺序，因此更充分地考虑了语义信息。实验结果表明，此算法虽然相对于 Word2vec 效果略有提升，但是对于文本相似性的度量效果依旧不佳。

综上所述，将来对于文本相似性度量的突破点在度量文本之间的语义相似性上，但是以上这些方法都忽略了文本之间语义相似性。Kusner、Sun 等人提出一种 WMD 文本相似度算法[60]。他们认为用单词来表征文本信息，丢失的文本信息更少，作者通过比较文本之间单词的相似性计算文本相似度。该算法的基本思想是在对文本中各个单词转变为词向量后，将文本 A 中各个单词的词向量转变为文本 B 中各个单词的词向量所需要花费的最小“功”，这个“功”就是这两个文本之间的相似度。WMD 算法基于 EMD（Earth Mover's Distance，搬土距离）算法通过单词向量之间的欧式距离以及各个单词的权重来计算文本之间的相似度。此算法忽略了文本内部语法逻辑，导致丢失了一些文本信息，特别是在计算长文本之间的相

似度时，效果表现不佳。

2.4 具有偏好次序的 0-1 稳定匹配算法

稳定匹配问题是运筹学中一个非常重要的问题，婚姻匹配问题是其中的一个典型问题。由于一个男生只能和一个女生相匹配，同时一个女生也只能和一个男生相匹配，所以婚姻匹配问题也被称为 0-1 稳定匹配问题。但是稳定匹配问题不可避免的会出现一部分元素匹配效果极好，另一部分元素匹配效果极差的情况。李巍等人提出了一种具有偏好次序的稳定匹配算法，能够在一定程度上实现全局最佳匹配[61]。

以婚姻匹配问题为例，在一个二部图中，二部图的一边 X 是男生集合，其中包含 4 个男生，二部图的另一边 Y 是女生集合，其中包含 4 个女生。对每一个男生而言需要在女生集合中寻找到一个最佳匹配对象，即：除了这个女生，他无法找到一个更好的匹配对象；对每一个女生而言需要在男生集合中寻找到一个最佳匹配对象，即：除了这个男生，她无法找到一个更好的匹配对象。X 中的男生 x_i 对 Y 中女生 y_j 的偏好序是 a_{ij}，Y 中女生 y_j 对 X 中男生 x_i 的偏好序为 b_{ji}，由于每一个人在自己心目中都存在一个配偶偏好排序，因此 a_{ij} 不一定等于 b_{ji}。如果将男生 x_i 和女生 y_j 配对，他们之间配对偏好关系为(a_{ij},b_{ji})，且称集合 X 与集合 Y 形成了一个匹配。与传统的稳定婚姻匹配算法不同的是，在这里不仅仅只考虑男生的心仪匹配对象 a_{ij}，还需要考虑女生的心仪匹配对象 b_{ji}，因此需要综合考虑男生和女生的整体想法，为所有人都找到一个满意的匹配对象。在一定规模的稳定婚姻匹配问题中，相同序的匹配效果优于不同序的匹配效果，因为这样避免了一方很满意，一方很不满意的情况，这对于不满意的一方是不公平的。例如：在规模为 10 的稳定婚姻匹配问题中，配对偏好关系为(5,5)相对于配对偏好关系为(4,6)或(3,7)的匹配关系更加稳定。因此定义匹配度 c_{ij}：

$$c_{ij} = (a_{ij}^{\ 2} + b_{ji}^{\ 2}) / (a_{ij} + b_{ji}) \tag{2-3}$$

c_{ij} 表示 x_i 和 y_j 之间的匹配度，在这里三者的匹配度分别为 5、5.2、5.8，c_{ij} 越小匹配关系越稳定。所以在考虑偏好次序的稳定婚姻匹配问题中，定义$((a_{ij},b_{ji}))_{n*n}$为集合 $X=\{x_1,x_2,...,x_n\}$与集合 $Y=\{y_1,y_2,...,y_n\}$之间元素偏好排名矩阵，c_{ij} 为配对(x_i ,y_j)之间的匹配度，该匹配度越小匹配关系越稳定。

由于一个人仅能和一个人相匹配，定义决策变量 $Z(i,j)$表示 x_i 和 y_j 相匹配。具有偏好次序的稳定匹配问题的数学模型为：

$$\begin{aligned} &\min \sum_{i=1}^{n}\sum_{j=1}^{m} Z(i,j) * c_{ij} \\ &s.t.: \sum_{j=1}^{m} Z(i,j) = 1 \\ &\sum_{i=1}^{m} Z(i,j) = 1 \\ &Z(i,j) \in \{0,1\} \end{aligned} \tag{2-4}$$

2.5 DTW 算法

DTW（Dynamic Time Warping，动态时间归整）算法是一种衡量时间序列相似性的算法，常用于数据挖掘和信息检索中。在计算两个非对齐或不等长的时间序列相似度时能够容忍一定程度的数据变形。

假设存在 $Q=\{q_1,q_2,...,q_n\}$和 $C=\{c_1,c_2,...,c_m\}$两个时间序列，其长度分别为 n 和 m。在计算时间序列相似度时，应该对时间序列按位比较，各个位越相似，则时间序列越相似。但是由于噪音和其他一些因素的影响，会导致相同的一段时间序列在不同状态下序列长度不相等，使得两个时间序列上相似节点产生错位，相似节点不会按位相似，而是在相近的位置上相似，例如：本应该 $q_1 \approx c_1$、$q_2 \approx c_2$ 等等，但是由于错位问题，会使得 $q_1 \approx c_2$、$q_2 \approx c_3$、$q_3 \approx c_4$ 等等。对于两个长度不相等的时间序列需要考虑对齐问题，所以也不能按位比较各时间节点的相似度大小。线性缩放是一种最简单的对齐方式，这种方式通过放大短序列使得长度等于长序列，或者缩短长序列使得长度等于短序列，从而实现长度对齐，基于此再进行序列相似度比较。但是现有的研究已经证明了这种方法效果不好。为了将序列对齐，构造一个维度为 $n*m$ 的矩阵 D，矩阵 D 中的位置(i, j)表示点 q_i 和 c_j 对齐，其中 $D(i, j)$对应的值表示 q_i 和 c_j 的距离。如果可以从 $D(1,1)$找到一条路径到达 $D(n,m)$，使得路径上值的和最小，那么相当于时间序列 Q 中的任意一个时间节点 q_i 都从时间序列 C 中找到了相对相似的时间节点，那么该最短路径和可以表示序列 Q 和序列 C 之间的相似度。因此问题可以转换为寻找一条连接网格中首末节点的最短路径和。

对于格点 $D(i,j)$而言具有以下三种性质：

（1）边界性，$1 \leqslant i \leqslant n$、$1 \leqslant j \leqslant m$；

（2）连续性，不可能跨过某个点进行匹配，只能和自己相邻的点对齐，保证

Q 和 C 中的每个坐标都能完成匹配。对于坐标点 $D(i,j)$的下一个坐标点 $D(i',j')$，i'-$i\leqslant 1$、j'-$j\leqslant 1$；

（3）单调性，匹配过程必须是随着时间单调进行的，因此对于坐标点 $D(i,\ j)$的下一个坐标点 $D(i',j')$，$0\leqslant i'$-i、$0\leqslant j'$-j。

累积距离 $r(i,j)$可以按下面的方式表示，累积距离 $r(i,j)$表示从起点到当前格点的最短距离。因此 $r(i,j)$可以表示为：

$$r(i,j)=D(i,j)+\min(r(i-1,j),r(i,j-1),r(i-1,j-1)) \tag{2-5}$$

最终得到的 $r(n,m)$即为最短距离，用它来表示时间序列之间的相似性。

2.6 本章小结

文本预处理是专利文本相似性分析的重要组成部分。本章首先介绍了文本预处理的相关技术，包括文本预处理的主要流程以及文本预处理的主要技术。然后介绍了自然语言处理中常用的 Word2vec 单词分布式表征技术。紧接着介绍了现有的一些文本相似度算法，同时阐述了传统算法存在的一些问题。最后介绍了稳定匹配算法和 DTW 算法，后文将基于它们进行专利文本相似性分析。

3　基于专利文本特点的实体识别和关键词位置编码

3.1　专利文本的特点

与普通文本相比，专利文本具有自身特点，主要体现在词和句式结构上。首先，对于专利文本中的“词”而言，文献[32]指出，专利文本中包含大量的合成词，这些合成词大多是专业术语，即：专利文本中的技术专有名词。技术专有名词在专利文本中有独特的语法结构，首先，大多是名词词性；其次，存在很多合成词。在前文中指出，通过挖掘专利文本的关键词可以高效的表征专利文本的语义信息。专利文本中的关键词不仅仅包括技术专有名词，还包括专利文本中联系技术专有名词之间关系的谓语动词。而技术专有名词又包括合成型技术专有名词和非合成型技术专有名词，对于合成型技术专有名词直接通过分词算法难以对其进行准确的切分，故而无法进行正确的识别，影响了关键词的提取结果。

其次，对专利文本中的句式结构而言，专利文本由许多陈述句构成。专利文本中不仅仅只包含结构为 SAO 句式的句子，通过后文对专利文本测量发现，专利文本以逗号、分号、句号为单位切分成“子句”集合后，所得到“子句”结构分为三种：“SAO 结构”、“SA 结构”、“AO 结构”。对于专利文本中形式为 SAO 句式、SA 句式、AO 句式的“子句”而言，与传统意义上的句子不同，传统意义上的句子一般以句号为单位，然而在专利文本中具有特殊形式的句子一般以逗号、分号、句号为单位进行切分才能得到，所以在专利文本中具有特殊形式的句子是一种特殊的“结构体”，本文称它为“子句”。以下将从“子句”构成模式、“子句”内部关键词和“子句”关系三个角度对专利文本中的 “子句”进行分析。

（1） “子句”的构成模式

对于专利文本中的三类“子句”而言，主语、谓语、宾语是“子句”中不同的句法成分。主语和宾语一般是名词，谓语一般是动词。由于停用词为连接词、介词、代词、标点符号、语气词，在去除专利文本停用词，保留专利文本关键词后，文本中剩下的单词大多数为名词词性和动词词性。这样，专利文本中的每个“子句”都可看成是由名词和动词组成的关键词集合。对于特定的句式结构，“子句”中的不同句法成分出现的位置有其基本规律，因此可以根据单词出现在“子句”中的位置区分出它的句法成分。本文认为出现在谓语动词之前的名词是“子句”的主语成分；出现在谓语动词之后的名词是“子句”的宾语成分。本文依托于分词工具的词性标注功能识别“子句”的关键词集合中各个单词的词性来识别该“子句”的句式结构。对于三类“子句”，它们的构成模式具有以下三种情况。

1）SAO 主谓宾结构：主谓宾结构是一种最常见的表达方式，主语一般用来说明句子中的人或事物，谓语一般用来说明主语的状态、特征或行为动作，宾语是主语通过动作行为所联系的对象[62]。对于该句式，“子句”中关键词词性的构成模式为若干个名词+动词+若干个名词。

2）SA 主谓结构：由一个或者若干个主语，加上一个或若干个谓语，所组成的句式[63]。对于该句式，“子句”中关键词词性的构成模式为若干个名词+动词。

3）AO 谓宾结构：它又称作动宾结构，动宾结构往往省略了前文出现的主语，它表示前文出现的主语对本句中的宾语所发生的支配关系和影响关系[64]。对于该句式，“子句”中关键词词性的构成模式为动词+若干个名词。

本文通过爬虫技术从知网中爬取了 35672 篇“电子”领域的专利数据，并提取出其中的 7918 篇专利构造测试集，爬取数据以及构造测试集会在第五章做详细介绍。对 7918 篇专利文本而言，总共包含 79432 个“子句”，它们的句式结构分布如下图 3-1 所示。从实验统计可知，对于构成这 7918 篇专利的“子句”而言，有大约 5%的“子句”无法区分它的基本句式。这种现象主要由于本文直接使用开源词性标注工具，它对于某些“子句”中的某些单词不能进行准确的词性标注。

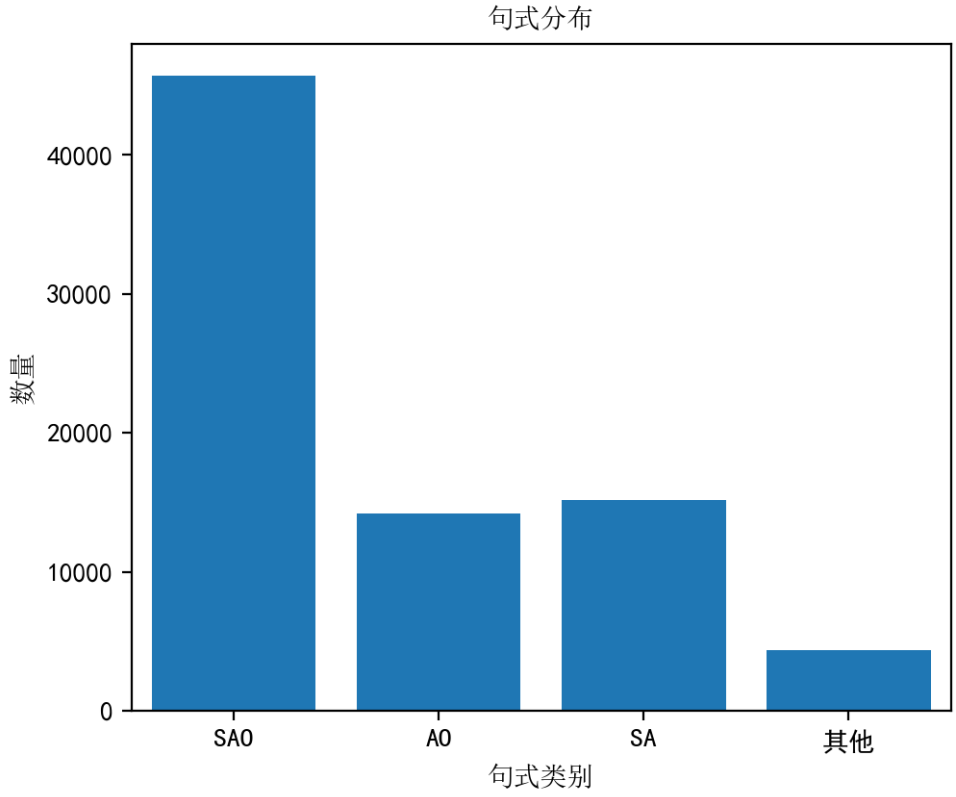

图 3-1　句式分布

Fig 3-1　Sentence pattern distribution

（2）“子句”内部关键词

1）“子句”中不同句法成分关键词对该“子句”的重要性程度不同

“子句”中不同句法成分的关键词对于“子句”的重要性程度不同，一般而言由于谓语动词一般是一些通用性动词（例如：“加上”、“覆盖于”等），而主语和宾语一般是技术专有名词，技术专有名词更能够表征专利文本的技术特点，所以主语和宾语对于专利文本“子句”的重要性程度更大。同时考虑到主语和宾语一般出现在“子句”的句首和句末，谓语一般出现在句中。因此，对于 SAO 句式的“子句”而言，句首和句末的关键词的重要性程度更高；对于 SA 句式的“子句”而言，句首的关键词的重要性程度更高；对于 AO 句式的“子句”而言，句末的关键词的重要性程度更高。

2）“子句”内部各个关键词之间的上下文关系

首先，对于“子句”中距离越近的关键词，上下文关系越强，关键词之间的关联程度越大。另外，在一个“SAO 子句”内部，宾语是主语通过动作行为所联系或支配的对象，所以主语和宾语之间通过谓语动词产生了相互依存的关系，在该“子句”特定的语境下，主语和宾语产生了一定的关联，虽然说主语一般出现在“子句”的靠前部分、宾语一般出现在“子句”的靠后部分，它们之间的位置较远，但是由于它们之间通过谓语动词产生了相互依存关系，所以在该“子句”特定的语境下，它们之间的上下文联系很紧密。例如：“稳定匹配算法可以应用于计算文本相似度”，“稳定匹配算法”和“文本相似度”在该子句下位置较远，但是在该子句特定的语境下“稳定匹配算法”和“文本相似度”通过谓语动词“应用”联系起来了，所以在本“子句”特定的语境下它们之间的关联程度较大。

（3）专利文本中“子句”之间的关系

由于语言的叙述有一定的逻辑性，所以文本中的各个“子句”具有紧密的前后文关系，在衡量专利文本相似性时，可以结合文本内部“子句”的上下文关系进一步充分利用专利文本的语义信息。

综上所述，在计算专利文本相似性时，将专利文本视为“子句”集合，不仅可以利用到专利文本关键词的语义信息，而且可以充分利用到“子句”的结构特点，因而可以更加充分的利用到专利的文本信息。因此这就对准确的识别专利文本中的关键词以及对“子句”内部关键词根据其上下文关系以及重要性进行定量处理提出了一定的要求。

本章主要分为两部分，第一部分首先介绍了准确提取专利文本关键词对于专利相似性度量的重要性，然后根据专利文本中关键词具有的特点对其进行识别和提取；第二部分首先介绍了专利文本中“子句”的句式特点，然后对专利文本中不同句式结构的 “子句”根据其结构特点对其中的关键词进行编码。

3.2　基于专利文本特点的合成型技术专有名词识别

3.2.1　问题描述

识别专利文本中的关键词对于计算专利文本相似性至关重要。然而专利文本中的关键词又有很大一部分是技术专有名词，这些技术专有名词构成了该专利的各方面属性，可以很大程度上表述该专利的技术特征。换句话说，如果成功识别出了专利文本中的技术专有名词，那么就能成功识别出专利文本中很大一部分的关键词。然而这些技术专有名词又有很大一部分是合成型技术专有名词。本文通过人工手动从文本中标注出 500 个合成型技术专有名词，利用 Jieba 分词对这些文本进行分词处理，通过人工分析这 500 个合成型技术专有名词，其中仅有 17%被正确切分，而字符数大于 4 的合成词完全没有被正确切分。由此可知，对于合成词，特别是长度较长的合成词难以被分词算法正确切分，如果不加以处理，直接将这些词作为专利文本关键词来表征文本的语义信息，势必会发生很多错误。因此，本节通过命名实体识别技术来提取专利中的合成型技术专有名词。

3.2.2　合成型技术专有名词的识别

在专利文本中存在大量的技术专有名词，这些专有名词又有很多是合成型技术专有名词，如果使用统计学的方法去识别专有名词，需要大量的标注语料，将专有名词从专利文本中标注出来，再利用模型进行训练，学习这种规律。但是现有的公开语料库又没有针对专利文本的标记语料。通过人工标记，费时费力。同时，专利文本中的技术专有名词具有其独特的构词方式，本文根据专利文本技术专有名词构词方式的特点，设计出匹配规则，使用基于规则的命名实体识别技术，以模式和字符串相匹配的方式，从专利文本中标注出技术专有名词。

Jieba 分词算法对专利文本中的非合成型实体词识别效果较好，但是对那些较长的合成型专有名词往往无法进行有效识别，常常会将一个合成型技术专有名词切分成多个词，例如：“水下航行器”会被切分成“水下”、“航行”、“器”，这是一种所处词+名词+名词的组合结构；“多量子阱层”会切分成“多”、“量子”、“阱层”，这是一种量词+名词+名词的组合结构；“固定槽”被切分成“固定”、“槽”，这是一种量形容词+名词性语素的组合结构。通常情况下，在一个句子中出现连续多个这样词性的词可能性不大，若出现，则有极大可能是分词算法将合成型技术专有名词切分成了多个词。

基于以上分析，本文依托词性标注工具，对 35672 篇专利文本中的各个词进行词性标注。通过统计高频出现的单词词性序列，然后再通过人工筛选的方式从中筛选出合成型技术专有名词的构词模式。通过构词模式和字符串相匹配的方式，可以从专利文本中标注出合成型技术专有名词。

对于专利文本中合成型技术专有名词总结的构词模式及其示例和该模式出现的频次如下表所示，汉语词性及其词性编码见附录中的中文词性表：

表 1　构词模式
Table 1　word-formation pattern

构词模式	频数	典型示例
s+n	867	水下|机器人、井底|钻头、井下|数据信号、野外|环境、室内|盆栽
s+n+n	320	水下|航行|器、海底|自行式|作业、井下|电磁|检测器
nz+n	6754	泵浦光|组件、多晶硅|还原炉、胺类|化合物、锂离子|电池
n+q	2584	外延|层、流线|腔、有源|层、半导体|层、电池|筒
n{2,}	83455	同位素|井间|液流、双信道|随钻|测量系统、尾座|支架|装置
eng+q	192	OLED|阵列、COA|阵列、PVC|片、InGaN|层、GaAs|层
v+q	1782	粘着|层、充电|桩、加固|杆、提升|座、承压|筒
a+n	6868	精密|成型、调谐|范围、正|六边形、固定|标气口、特殊|材料
eng+k+n+g+q	69	P|型|氮化|镓|层、P|型|氮化|镓|系列、N|型|氮化|镓|层
n+g+q	130	量子|阱|层、氮化|镓|层、高温|阱|层、大豆|甙|元、苯|脲|类
n+g	1098	量子|阱、氮化|镓、芒果|蒂、芳基|蒽、钨酸|铋、金属|钯、有机|铋
eng+k+eng+q	69	p|型|GaN|层、n|型|GaN|层、P|型|AlGaN|层、P|型|InGaN|层
n{2,}+nz	1386	文本|信息、电力|材料|防腐剂、信号|接收器、电磁|随钻|测量系统
eng+vn+n	470	多|量子|阱层、FM|调制|发送器、SERS|检测|装置、MOS|驱动|电路
eng+n+n	2887	多|量子|阱层、MOS|驱动|电路、TOFD|监测|钢板、DNA|检测|方案
vn+n	11719	辅助|平台、生理|指标、运动|组件、驱动|机构、调节|系统
eng+l	237	AC|逆变器、WiFi|微处理器、AC|自锁、PLC|控制器、GSM|数据传输
f+n+v	1952	外部|液体|流入、外部|传感|器、外|侧壁|位置、两侧|光栅|占空比
n{1,}+q	3330	软性|基材|层、防震|层、螺纹|杆、电路|腔、器材|盒
b+nz	678	抗|压、凹|槽、联轴|器、滤|膜、测量|杆、二级|泵浦光、多模|泵浦
b+n	10105	重|金属、支撑|架、运算|服务器、液压|缸、升降|舵
a+n	6868	高|密封性、低|精度、显著|优点、滑套|信息、干燥|污泥

以该表第一行为例，本文对该表做一些简单的解释：对 35672 篇专利文本中出现 s（方位词）+n（名词）的连续单词序列的次数为 867 次，它所对应的一些连

续单词序列示例包括：“水下”+“机器人”、“井底”+“钻头”、“水下”+“数据信号”、“野外”+“环境”、“室内”+“盆栽”。可以看出对于以上连续单词序列可以视为一个完整的单词，这些单词都是合成型技术专有名词。

在匹配过程中本文采取正向最大匹配法，如果在文本中既存在能和 n+n+n 匹配的字符串 1，又存在能和 n+n 匹配的字符串 2，同时字符串 2 是字符串 1 的子串，本文选择最大匹配 n+n+n 所对应的字符串 1 作为识别出来的合成型技术专有名词。例如：若文本中存在“同位素井间液流”，该字符串为 n+n+n 构词模式，所以它既能被 n+n 构词模式识别，又能被 n+n+n 构词模式识别，如果在这里将它视为 n+n 构词模式，那么对于文本中的“同位素井间液流”，“同位素”和“井间”将被合并识别为“同位素井间”，因此“同位素井间液流”将被重新识别为“同位素井间”和“液流”，在这里它们又同为名词词性，在文本中表现为 n+n 序列，然而文本中出现连续两个名词词性的词可能性不大，依旧不符合正常的表达习惯，因此在这里正确的匹配模式是 n+n+n。

但是基于规则的命名实体识别技术存在一些缺点：

（1）无法穷尽专利文本中的合成型技术专有名词的构词规律。尽管可能无法对专利文本中的合成型技术专有名词完全识别，但是可以在一定程度上优化技术专有名词的识别，能够提取出更准确的词来表征专利文本的语义信息，这对专利文本相似度的度量将有一定的帮助。通过实验，针对 35672 篇专利文本，利用以上规则，共识别出 123939 个合成型技术专有名词。

（2）会造成一些误识别，可能会提取出一些并不是技术专有名词的合成词，甚至所提出来的字符串可能都不是一个词。

通过人工验证的方式，随机提取 1000 个识别出来的技术专有名词进行验证，仅有 21 个词存在上述误识别现象，占验证数据的 2.1%。

3.3　基于专利句式结构的关键词位置编码

3.3.1　问题描述

在计算专利文本相似性时，为了更加充分的利用专利的文本信息，应该不仅仅利用到专利文本中关键词的语义信息，还应该考虑到专利文本的结构特点。本小节针对 3.1 节对专利文本句式特点的分析，提出了一种对专利文本各“子句”中关键词的位置编码方式。通过该编码方式，不仅可以区分“子句”中关键词的上下文关系，还可以区分关键词对该“子句”的重要性程度。

3.3.2　基于专利文本句式结构的关键词位置编码

如 3.1 节所述，“子句”中不同的关键词对于“子句”的重要性程度不同，一般而言“子句”中的技术专有名词更能够表征“子句”的关键信息，技术专有名词一般出现在 “子句”中的主语和宾语部分，因此主语和宾语相对于谓语更能够表征“子句”的关键信息。本文定义：编码值越小的关键词，对于“子句”越重要，因此“子句”中主语和宾语对应关键词的位置编码值相对于谓语的位置编码值更小。

除此以外，在自然语言处理中，句子内部的单词顺序是很重要的特征[65]，它不仅可以在一定程度上表征关键词的句法成分信息，还可以在一定程度上表征关键词的上下文信息。因而在通过对“子句”中的关键词进行位置编码时，还需要考虑到关键词出现在“子句”中的顺序。

首先，对于“子句”中距离越近的关键词，上下文关系越强，关键词之间的关联程度越大。因此本文定义：“子句”中位置越近的关键词，位置编码值的差值越小。

另外，由前文可知，在一个“SAO 子句”内部，宾语是主语通过动作行为所联系或支配的对象，所以主语和宾语之间通过谓语动词产生了相互依存的关系，在该“子句”特定的语境下，主语和宾语产生了一定的关联，虽然说主语一般出现在“子句”的靠前部分、宾语一般出现在“子句”的靠后部分，它们之间的位置较远，但是由于它们之间通过谓语动词产生了相互依存关系，所以在该“子句”特定的语境下，它们之间的上下文联系很紧密。因此“子句”中的主语和宾语对应的关键词编码值也应该相差不大。

基于上述专利文本的句式结构特征分析，本小节对句式为 SAO、SA、AO 的三类句式分别进行以下位置编码，对无法准确识别句式的“子句”本文将其视为“SAO 子句”。假设子句 S 共包含 l 个单词，其中的第 k 个单词 S_k 的位置编码为 P_{S_k}。

（1）SAO 子句

如上文所述，考虑到“子句”内部各个关键词之间存在上下文关系，距离越近的关键词，上下文关系越强，因此“子句”间位置越近的关键词，位置编码的差值越小；同时，在该句式下，主谓宾句法成分完整，主语和宾语之间存在依存关系，而主语一般出现在“子句”中靠前部分、宾语一般出现在“子句”中靠后部分，因此认为在 SAO“子句”中靠前和靠后的单词之间具有一定的相互依存关系，它们之间的上下文联系紧密，编码值也应该相近；另外，对于专利文本“子句”中的关键词，主语关键词和宾语关键词相对于谓语关键词更能够表征该“子句”的关键信息，而在前文中提到，编码值越小的单词相对于专利“子句”越重

要。所以对该句式中的关键词采取一种“Λ 型”对称编码方式，它是从句首到句末先递增再递减的一种编码方式，编码差值越小的关键词说明它们之间的紧密性程度更大，同时关键词编码值越小说明该关键词对于“子句”的重要性程度越高。

$$P_{S_k}=\begin{cases}\dfrac{k}{(1+l/2)*(l/2)} & l\text{被2整除}and\ k\le l/2\\ \dfrac{l-k+1}{(1+l/2)*(l/2)} & l\text{被2整除}and\ k>l/2\\ \dfrac{k}{(1+\lfloor l/2\rfloor)*\lfloor l/2\rfloor+(\lfloor l/2\rfloor+1)} & l\text{不被2整除}and\ k\le\lfloor l/2\rfloor\\ \dfrac{l-k}{(1+\lfloor l/2\rfloor)*\lfloor l/2\rfloor+(\lfloor l/2\rfloor+1)} & l\text{不被2整除}and\ k>\lfloor l/2\rfloor\end{cases} \tag{3-1}$$

当一个“SAO 子句”的长度为 4 时，通过本编码方式，该“子句”每个位置的单词的编码分别为：1/6、2/6、2/6、1/6。很明显，通过这样的编码方式，不仅使得“子句”内部相近的关键词编码值相近，还使得在“子句”内部靠前和靠后的可能存在相互依存关系的关键词编码值相近，同时还使得主语/宾语关键词的位置编码值较小、谓语关键词的位置编码值较大，更能够体现出不同的关键词对“子句”的重要性程度。

（2）SA 子句

在该句式下，由于“子句”中只包含主语和谓语成分，不存在宾语成分，所以该“子句”的句末相当于主谓宾句式的句中。本文采取一种从句首到句末线性递增的编码方式，编码值越小说明该关键词对于“子句”的重要性程度越高。很明显，通过这样的编码方式，不仅使得主语对应的关键词编码值更小，还使得“子句”内部相近的关键词编码值相近。

$$P_{S_k}=\frac{k+1}{((1+l)*l)/2} \tag{3-2}$$

当一个“SA 子句”的长度为 3 时，通过本编码方式，该“子句”每个位置的关键词的编码分别为：1/6、2/6、3/6。

（3）AO 子句

在该句式下，由于“子句”中只包含谓语和宾语成分，不存在主语成分，所以该“子句”的句首相当于主谓宾句式的句中。所以本文采取一种从句首到句末

线性递减的编码方式，编码值越小说明该关键词对于“子句”的重要性程度越高。很明显，通过这样的编码方式，不仅使得主语对应的关键词编码值更小，还使得“子句”内部相近的关键词编码值相近。

$$P_{S_k} = \frac{l-k}{\left((1+l)*l\right)/2} \tag{3-3}$$

当一个“AO 子句”的长度为 3 时，通过本编码方式，该文本每个位置的关键词的编码分别为： 3/6、2/6、1/6。

针对不同的句式，分别使用上述编码方式进行“子句”内部的位置编码。通过上述编码方式，可以有效区分关键词的上下文关系以及关键词的重要性程度。

3.4 本章小结

准确提取出专利文本中的关键词能够高效的表征文本的语义信息，通过比较专利文本之间关键词的相似性能够高效的度量专利文本相似性，因此正确识别专利文本中的关键词对于度量文本相似性至关重要。同时专利文本具有特定的句式结构，如果在比较专利文本之间的相似性时，综合考虑专利文本的句法特点，可以更加充分利用专利的文本信息。

本章首先介绍了命名实体识别对于专利文本相似性度量的重要性，然后基于专利文本中合成型技术专有名词的构词方式，提出一种基于规则的命名实体识别方法，优化专利文本中技术专有名词的识别。最后，为了充分利用专利文本的句式结构特点，提出了一种对专利文本“子句”中关键词的位置编码方式。

4 基于句法表征的专利文本相似度算法

4.1 问题描述

专利文本之间的相似度可以通过比较构成文本的元素来度量，如果构成两篇文本的元素越相似，则这两篇文本越相似。考虑到文本是由“子句”构成，一个“子句”包含文本中的部分语义信息，如果两篇文本相似，则势必会存在语义相似的“子句”；同时“子句”是由关键词构成，如果两个“子句”相似，则势必会存在语义相似的关键词。因此，可以首先通过专利文本“子句”中各个关键词之间的相似度计算“子句”之间的相似度；然后通过专利文本“子句”之间的相似度计算专利文本相似度。

然而文本也可以看作以关键词构成的，为什么要先通过文本“子句”中关键词之间的相似度计算专利文本“子句”之间的相似度，然后通过“子句”之间的相似度计算文本之间的相似度，而不是直接通过文本关键词之间的相似度计算文本之间的相似度呢？主要基于以下四点观察：

第一，如果孤立且分散的考虑两篇长文本内部各个关键词之间的相似度来计算文本之间的相似度，会忽略文本中每个词的上下文信息这一重要语言成分[66]；

第二，以“子句”为单位时，由于“子句”内部各关键词存在极强的上下文关系，利用单词计算“子句”相似度时，通过综合比较“子句”内部各个关键词之间的相似性得到的“子句”相似性，自然而然的利用到了关键词的上下文信息[66]；

第三，专利文本中的“子句”具有 SAO、SA、AO 句式结构，所以“子句”中的关键词都具有特定的句法成分信息（主语、谓语、宾语）。不同“句法成分”对“子句”的重要性程度不同。如果存在两个相似专利，那么它们之间一定存在相似的语义信息，因此存在语义表达相似的“子句”。而“子句”中不同句法成分对“子句”的重要性程度不同，如果两个“子句”越重要的“关键词”越相似，则这两个“子句”越相似。所以在以“子句”为单位计算文本相似度时，可以不仅仅只利用文本中关键词的语义信息，还可以利用到关键词的句法成分信息；

第四，在得到两篇文本各个“子句”间的相似度后，对于一个“子句”而言，它表征了专利文本一部分的语义信息，专利文本各个“子句”之间的相似度等价于专利文本各部分语义信息之间的相似度。由于文本中语义的表达有一定的逻辑，具体表现在：文本中前后“子句”存在关联，所以可以将文本“子句”序列视为时间序列，通过比较文本“子句”序列之间的相似性可以度量文本之间的相似性。

对于上述的第一点、第二点、第三点特征，通过 3.3.1 节所述的关键词位置编

码能够进行有效的定量处理。基于上述分析，本章提出一种专利文本相似度算法，称它为 SRMA 算法（Syntactic Representation Matching Algorithms，句法表征匹配算法）。主要分为两部分：第一部分为通过专利文本“子句”之间各个关键词的语义信息和位置信息来计算专利文本“子句”之间的相似度；第二部分为根据专利文本各个“子句”之间的相似度计算专利文本之间的相似度。在计算专利文本 *a* 和专利文本 *b* 相似度时，首先将它们拆分成“子句”序列，各个“子句”同时也表示为一个关键词序列，提取每个关键词的语义信息和位置信息，利用子句中关键词的语义信息和位置信息计算专利文本 *a* 和专利文本 *b* 间各个“子句”的相似度；然后根据专利文本 *a* 和专利文本 *b* 间各个“子句”的相似度和“子句”的位置信息计算文本之间的相似度，文本相似度算法流程图如图 4-1 所示。

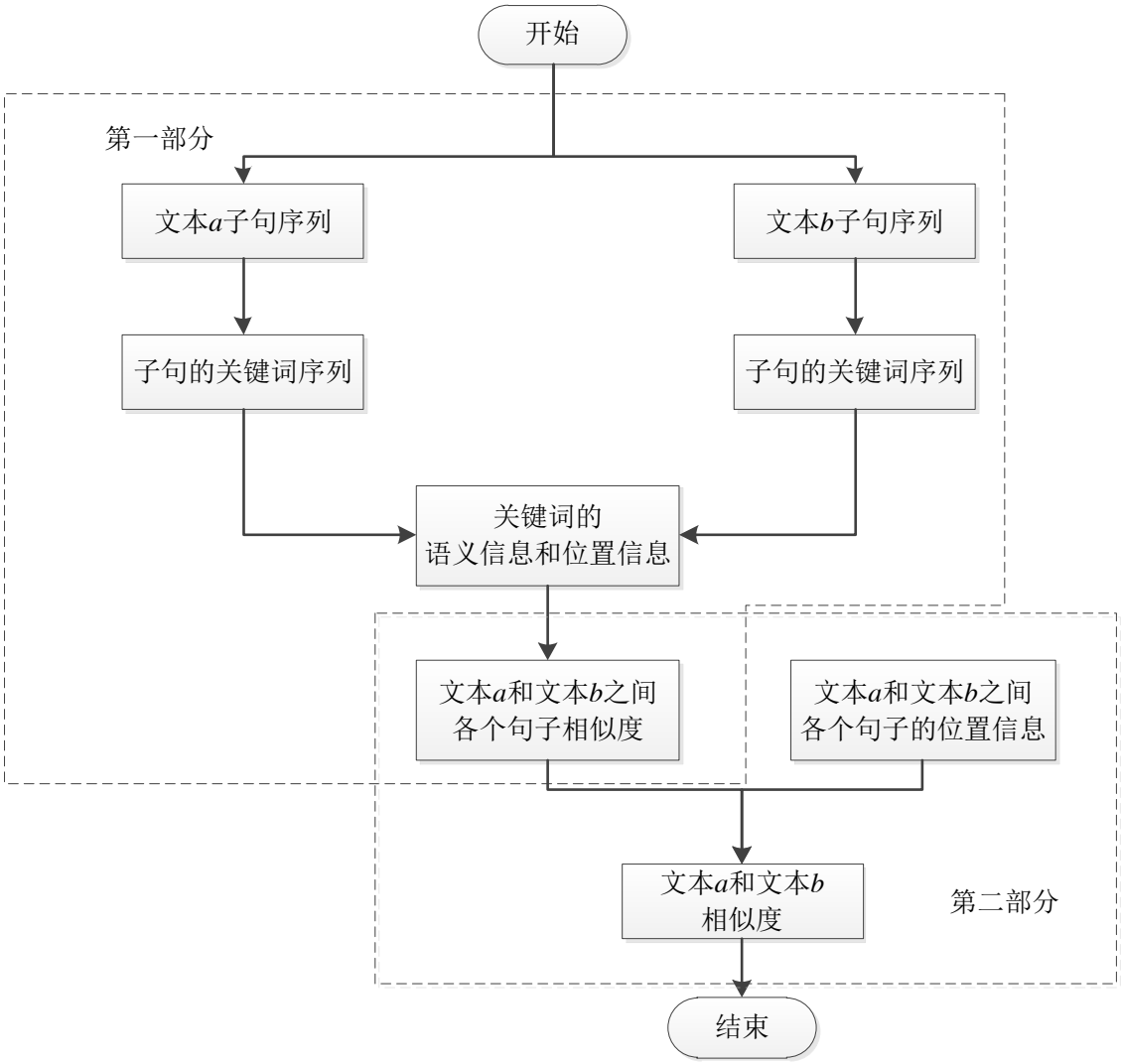

图 4-1　文本相似计算流程
Fig 4-1　Text Similarity Computing Process

4.2 专利子句之间的相似度计算

4.2.1 基本思想

“子句”是以关键词组成的。从第三章可知，“子句”中关键词的位置信息可以在一定程度上表现关键词的上下文信息和关键词的重要性程度。对于具有相似语义的两个“子句”而言，它们之间不仅仅应该存在相似语义的关键词，而且这些关键词对各自“子句”的重要性程度和这些关键词出现在各自“子句”中的“位置”都应该相似。

后文将基于“子句”之间关键词的位置编码和关键词的语义信息提出了一种专利文本“子句”相似度的计算方法。

4.2.2 算法设计

在通过关键词计算“子句”相似度时，如果两个“子句”间各个位置编码相近的关键词语义相似，则说明在考虑“子句”内部关键词上下文关系和句法成分信息的前提下，如果两个“子句”之间重要性程度相近的关键词语义也相似，那么两个“子句”越相似。故可以以关键词为粒度，通过比较“子句”之间各个关键词的语义相似性和位置信息来度量专利文本“子句”相似性。假设待计算相似度的两个“子句”分别表示为集合 $X=\{x_1,x_2,...,x_n\}$ 和集合 $Y=\{y_1,y_2,...,y_m\}$，x_i 为“子句 X”中的一个关键词，y_j 为“子句 Y”中的一个关键词。定义关键词 x_i 和关键词 y_j 的相似度为：

$$S_{ij}=\left|P_{x_i}-P_{y_j}\right|*d_{ij} \tag{4-1}$$

在这里$|P_{xi}-P_{yj}|$为待匹配的两个关键词之间的位置相似度，d_{ij} 表示关键词 x_i 和关键词 y_j 之间的语义相似度。

在度量专利“子句”之间的相似度时，往往两个“子句”之间相似的关键词越多，则这两个“子句”之间的相似度越高。同时对于“子句”中的任意一个关键词，它对于“子句”而言都存在一定的位置信息（位置信息又包含关键词的上下文信息和关键词的句法成分信息），且它表述了“子句”的部分语义信息。对构成“子句”的不同关键词而言，由于它们表述的语义信息和位置信息不可能完全相同，所以构成“子句”的关键词之间存在差异，对于“子句”而言每个关键词都存在无法替代的作用，因此在利用关键词相似度计算“子句”相似度时，“子

句”中的一个关键词只能和另一个“子句”中的一个关键词相匹配。所以，当能够在两个“子句”之间为每个关键词找到一个稳定匹配关键词，若各个匹配关键词越相似，则“子句”越相似。同时与具有偏好序的稳定匹配问题类似的是，对于关键词 x_i 而言“子句 Y”中与其最相似的关键词是 y_j，但是对于关键词 y_j 而言“子句 X”中与其最相似的关键词不一定是 x_i，在这里存在偏好序问题。故该匹配关系满足 2.4 节所述的具有偏好序的 0-1 稳定匹配[61]。

如 2.4 小节所述，关键词 x_i 和关键词 y_j 之间的相似度偏好序为(a_{ij},b_{ji})，全局的最佳匹配关系需要统筹考虑二部图两边各个关键词之间的相似度偏好序，因此关键词之间的全局匹配系数 c_{ij} 表示为：

$$c_{ij}=\left(a_{ij}^{\ 2}+b_{ji}^{\ 2}\right)/\left(a_{ij}+b_{ji}\right) \tag{4-2}$$

在整体范围内为每个关键词找到一个全局最稳定的匹配关键词，匹配关系表示为：

$$Z=\left(Z(i,j)\right)_{n*m}，\ Z(i,j)=0或1 \tag{4-3}$$

该匹配关系表示为：在全局范围内，对“子句”中每一个关键词都只能从另一个“子句”中找到一个最合适的关键词进行匹配，使得全局的匹配系数和最低。因此，具有偏好序的“子句”关键词稳定匹配算法表示为：

$$\begin{aligned}&\min \sum_{i=1}^{n}\sum_{j=1}^{m}c_{ij}*Z(i,j)\\ &s.t.:\ \sum_{j=1}^{m}Z(i,j)=1\\ &\sum_{i=1}^{n}Z(i,j)=1\\ &Z(i,j)\in\{0,1\}\end{aligned} \tag{4-4}$$

“子句”间关键词越相似，两个“子句”越相似。因此“子句”之间的语义相似度表示为匹配关键词之间的相似度之和：

$$sim(\mathrm{X,Y})=\sum_{i=1}^{n}\sum_{j=1}^{m}S_{ij}*Z(i,j) \tag{4-5}$$

当两个“子句”词数不相等时，无法为其中一个“子句”中的关键词都在另一个“子句”中找到配对。本文认为在当前部分匹配关系下求出来的是两个“子

句”在一定语义占比下的相似度。当两个“子句”匹配词数越多，则匹配上的语义信息越多，语义占比表示为：

$$p = min\left(len\left(s_{x_i}\right), len\left(s_{y_j}\right)\right) / max\left(len\left(s_{x_i}\right), len\left(s_{y_j}\right)\right) \tag{4-6}$$

若两句话在越大语义占比上相似，则两个“子句”越相似。因此“子句”之间的相似度表示为：

$$sim(\mathrm{X},\mathrm{Y}) = \sum_{i=1}^{n}\sum_{j=1}^{m} S_{ij} * Z(i,j)/p \tag{4-7}$$

如图 4-2 所示，在两篇专利文本中存在子句 $X=(x_1,x_2,x_3,x_4)$和子句 $Y=(y_1,y_2,y_3,y_4)$，如果两个句子中每个关键词根据 0-1 稳定婚姻匹配算法得到如图所示的匹配结果后，“子句 X”和“子句 Y”之间的相似度就是匹配关键词之间的语义相似度之和：

$$sim(X,Y) = S_{11} + S_{24} + S_{32} + S_{43} \tag{4-8}$$

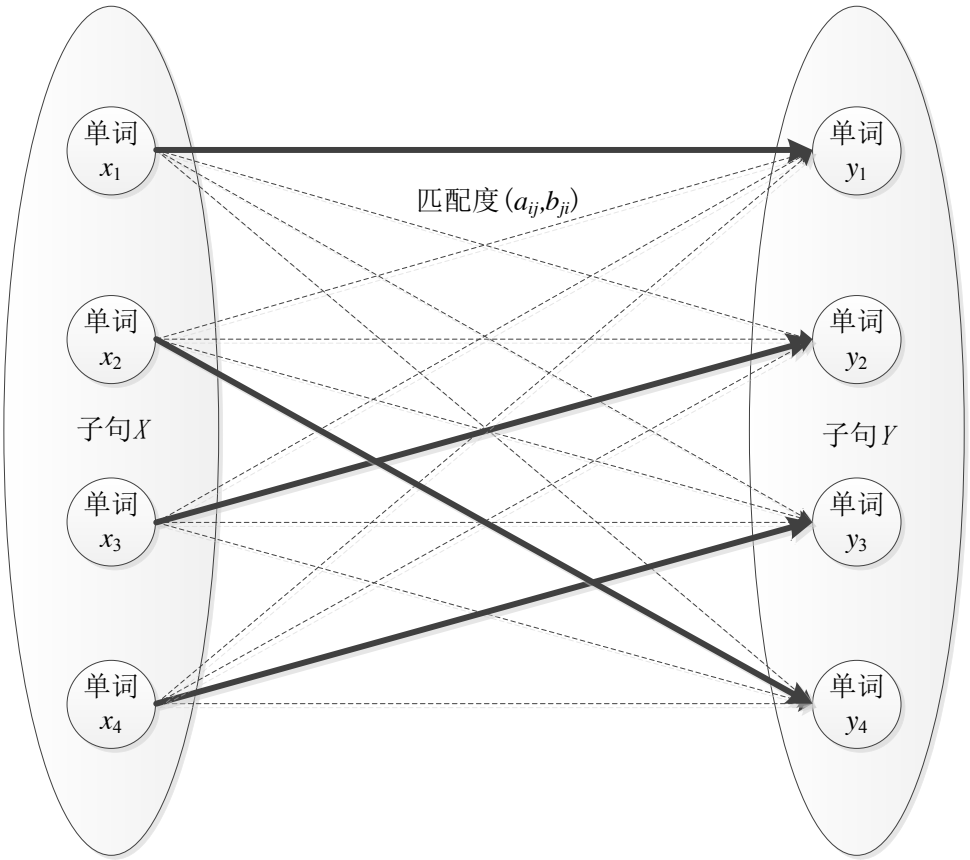

图 4-2　子句之间关键词的稳定匹配
Fig 4-2　Stable matching of words between sentences

4.3 专利文本之间的相似度计算

4.3.1 基本思想

文本是以“子句”组成的，对于两篇文本而言，如果它们相对应的“子句”越相似，则这两篇文本越相似。同时，对于一篇文本而言，其中的各个“子句”具有前后文关系，所以文本可以看作以“子句”为基本单元组成的时间序列集合，将文本中第 i 个“子句”视为时间序列集合中第 i 个时刻对应的元素。所以在计算专利文本之间的相似度时，可以利用 DTW 算法将其转换为计算两个时间序列之间的相似性。

利用 DTW 算法计算专利文本相似性的优势在于：

（1）通过文本“子句”相似性计算文本相似性的过程与计算时间序列相似性完全相同。在计算时间序列相似性时，两个完全相同的时间序列对应时刻的状态应该完全相同，对于两篇完全相同的文本而言，它们之间相同序号的“子句”应该完全相同。

（2）DTW 算法可以用于计算长度不相等的时间序列相似度，也可以用于计算相似状态有一定偏差的时间序列相似度，这点也与文本相似度计算过程相似。“子句”在文本中出现的顺序与专利文本所描述的对象有关，对于相似专利而言，所描述的对象相同，因此行文顺序也有一定的相似，所以相似专利中语义相似的“子句”序号相近，例如：存在相似专利 A 和专利 B，专利 A 的第一个“子句”和专利 B 的第一个“子句”相似、专利 A 的第二个“子句”和专利 B 的第二个“子句”相似、专利 A 的第三个“子句”和专利 B 的第三个“子句”相似，依此类推。然而对于相似专利文本而言，写作风格不一定相同且包含的子句数量不一定相同，所以它们之间相似“子句”的序号可能会略有偏差，但是由于所描述的对象相同，行文顺序有一定相似性，所以偏差不会太大，例如：专利 A 中第一个“子句”和专利 B 中第二个“子句”最相似，专利 A 中第二个“子句”和专利 B 中第三个“子句”最相似。

4.3.2 算法设计

假设待计算相似度的两个文本分别表示为集合 $X=\{x_1,x_2,...,x_n\}$ 和集合 $Y=\{y_1,y_2,...,y_m\}$，x_i 为文本 X 中的一个“子句”，y_j 为文本 Y 中的一个“子句”。

如 4.3.1 小节所述，文本“子句”序列可以视为一个时间序列集合，例如：文本 X 中“子句 x_i”视为时间序列 X 中第 i 个时刻点的状态，文本 Y 中“子句 y_j”视为时间序列 Y 中第 j 个时刻点的状态。因此，文本 X、Y 之间的相似性可以转换为

时间序列 X、Y 之间的相似性。由于文本 X、Y 之间各个“子句”的相似度在 4.2 节已经得到，所以可以得到相似度矩阵 D：

$$D=\left(D(i,j)\right)_{n*m} \tag{4-9}$$

其中 $D(i,j)$表示“子句 x_i”和“子句 y_j”之间的相似度。

根据 2.5 小节所述，时间序列 X、Y 之间的相似度可以转换为从 $D(1,1)$到 $D(n,m)$之间的最小累计距离，定义为 $r(n,m)$。其中 $r(i,j)$ 表示为从 $D(1,1)$到 $D(i,j)$的最小累计距离，$r(i,j)$需要满足以下条件：

$$r(i,j)=D(i,j)+\min(r(i-1,j),r(i,j-1),r(i-1,j-1)) \quad (1\le i\le n,1\le j\le m) \tag{4-10}$$

文本 X、 Y 之间的相似度可以转换为时间序列 X、Y 之间的相似性。由 DTW 算法可知，时间序列 X、Y 之间的相似性可以表示为从 $D(1,1)$到 $D(n,m)$之间的最小累计距离，即：

$$sim(X,Y)=r(n,m) \tag{4-11}$$

因此文本 X、 Y 之间的相似度等于 $r(n,m)$。

4.4 本章小结

相对于现有的一些技术，它们通常局限于关键词的语义信息对文本相似性的影响，为了更加充分的利用专利文本中关键词的语义信息和句式结构，本章提出了一种基于句法表征的专利文本相似性度量算法。通过将文本切分成“子句”集合，通过综合比较“子句”内部关键词的位置信息和关键词的语义信息来度量“子句”之间的相似性，这样一来在计算文本相似性时考虑了更加丰富的文本信息。然后将一篇文本视为一个时间序列集合，利用 DTW 算法基于专利文本“子句”相似度计算专利文本之间的相似度。

5 算法的实现和验证

5.1 实验环境

为了加快计算机运行速度和数据文件的加载速度，本论文的仿真实验使用固态硬盘用来存储相关数据文件。由于 Python 编程语言包含丰富机器学习、深度学习函数库，例如：sklearn、gensim，所以本文使用 Python 进行实验。

本文所有的实验基于以下软硬件配置：

（1） 处理器：3.1GHz 双核 Intel Core i5 处理器

（2） 内存（RAM）：8G

（3） 硬盘：SSD 256G

（4） 系统类型：MacOS

（5） 软件环境：JetBrains PyCharm Community Edition 2018.2.2 x64

（6） 编程语言：Python

（7） 网络环境：校园网

5.2 整体流程

专利相似度计算的基本流程如图 5-1 所示，其基本过程如下：

首先，利用爬虫技术爬取知网，收集专利的相关数据，包括：专利的标题、专利的摘要、专利的主权项、专利的主分类号、该专利的相似专利，并对专利继续南行切分构造专利测试集和专利数据库；其次，在获得专利数据后，为了使计算机能够识别并处理文本信息，对专利标题和专利主权项数据进行数据预处理，预处理过程主要包括词性标注、合成型技术专有名词识别、中文分词、停用词处理、关键词提取；然后，对提取的文本关键词集合利用 Word2vec 模型训练词向量表征；最后，通过本文设计的基于句法表征的专利文本相似度算法计算专利测试集中各个专利和专利库中各个专利之间的相似度，并从专利库中提取出最相似的五个专利作为测试专利的相似专利，最后进行算法评估。

根据具体的流程，本文把专利相似度计算分为 5 个步骤，如图 5-2 所示。其中步骤 A 为知网数据采集；步骤 B 为专利文本预处理；步骤 C 为构建待查找的专利测试集；步骤 D 为构建专利数据库；步骤 E 为专利相似度计算，并根据测试专利与专利数据库中各个专利的相似度，推荐出最相似的五个专利。

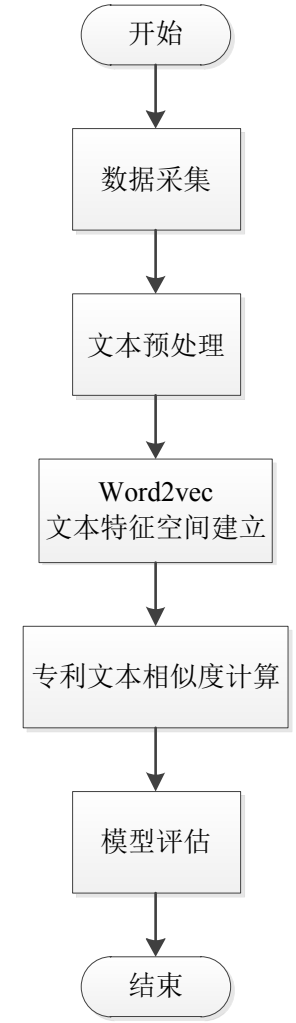

图 5-1 流程图
Fig 5-1 Flow Chart

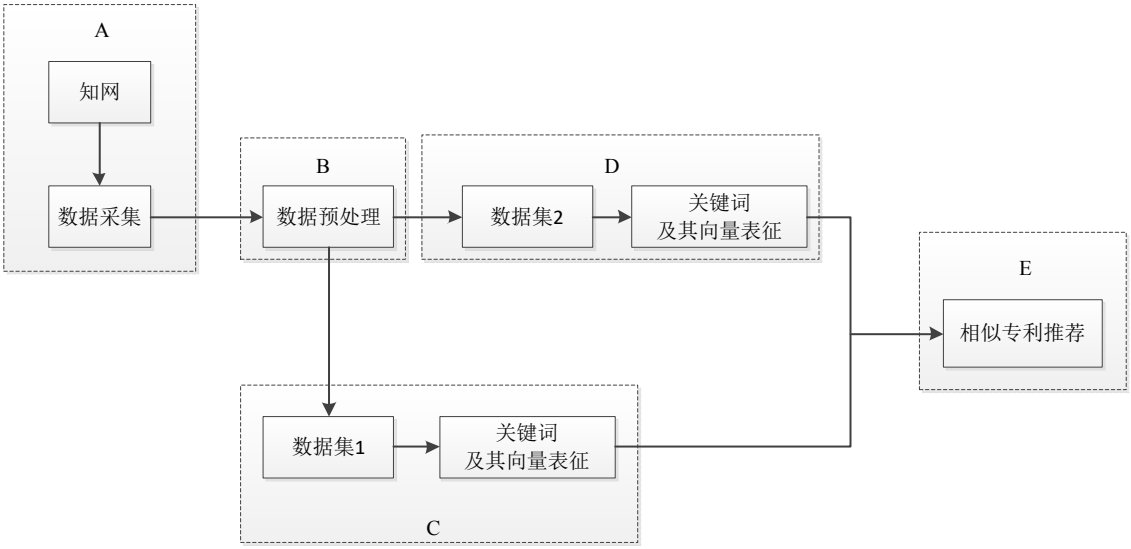

图 5-2 步骤图
Fig 5-2 step diagram

第一部分为数据采集, 仅包括步骤 A。本文使用爬虫技术完成了对知网上电子领域的专利数据采集工作。并将专利数据分为两个部分：数据集 1 是待查找专利数据集；数据集 2 是构建的专利数据库。

第二部分为文本预处理，包括步骤 B、步骤 C、步骤 D。文本预处理的步骤是：词性标注、合成型技术专有名词识别、中文分词、停用词去除、关键词提取。在对专利文本进行中文分词及词性标注时，本文采用的是 Jieba 分词工具，并根据合成型技术专有名词的构词规则，利用第三章提出的基于构词规则的命名实体识别算法识别出专利数据集中合成型技术专有名词。停用词去除部分是基于人工构建的停用词表，通过字符串匹配的方式，删除文本中出现在停用词表中的单词。考虑到专利文本中的关键词多为名词词性或动词词性，故提取词性为名词和动词词性的单词作为专利文本中的关键词。

第三部分为专利相似度计算，仅包括步骤 E。为待测试的专利数据集中的每个专利在专利数据库中查找与它们最相似的五个专利，并比较知网给出的相似专利，评估算法效果。

5.3 数据采集与处理

5.3.1 数据采集

本文使用爬虫技术在知网上以“电子”为关键词，完成主题为“电子”的专利数据采集工作。知网上相关页面如下图 5-3 所示。基于 selenium 库实现了知网专利数据的提取，本文提取出了其中的专利标题、摘要、主权项、主分类号、专利分类号、该专利的相似专利，其中专利标题和专利主权项是后文度量专利相似性的主要依据。我们需要从构建的专利数据库中找到专利测试集中各个专利的相似专利，专利测试集中包含 529 个专利；专利数据库中包含 7389 个专利，它们中包含待查找专利数据的所有相似专利。

专利测试集以及专利数据库中的文本数据以逗号、分号、句号切割而成的“子句”数量分布图如图 5-4 所示。该图的横坐标是专利文本中包含“子句”的数量 x，纵坐标是专利文本的数量 y，该图表示有 y 个专利文本包含 x 个“子句”。由此图可知大多数专利文本包含的“子句”个数都在 5-15 个之间，而“子句”个数超过 40 的文本几乎没有，文本中的“子句”数量分布较为均匀。

一个专利文本包含一个主分类号，主分类号能代表本专利所属技术领域。利用专利主分类号来可以实现对专利测试集以及专利数据库中的专利文本数据进行技术领域的划分，专利测试集以及专利库中的 7918 个专利文本属于 494 个不同主

分类号，因此专利文本属于 494 个不同技术领域。对应每一个技术领域下都包含一定数量的专利文本。如图 5-5 所示，表示各个主分类号下专利文本数量的直方图。由图可知，大多数技术领域下的文本数量在 5-15 之间，由此可见，本文构造的专利库中不同技术领域下的文本数量分布均匀。因此，本文已经在一定程度上模仿真实专利数据库了。

图 5-3 知网示意图

Fig 5-3 HowNet schematic diagram

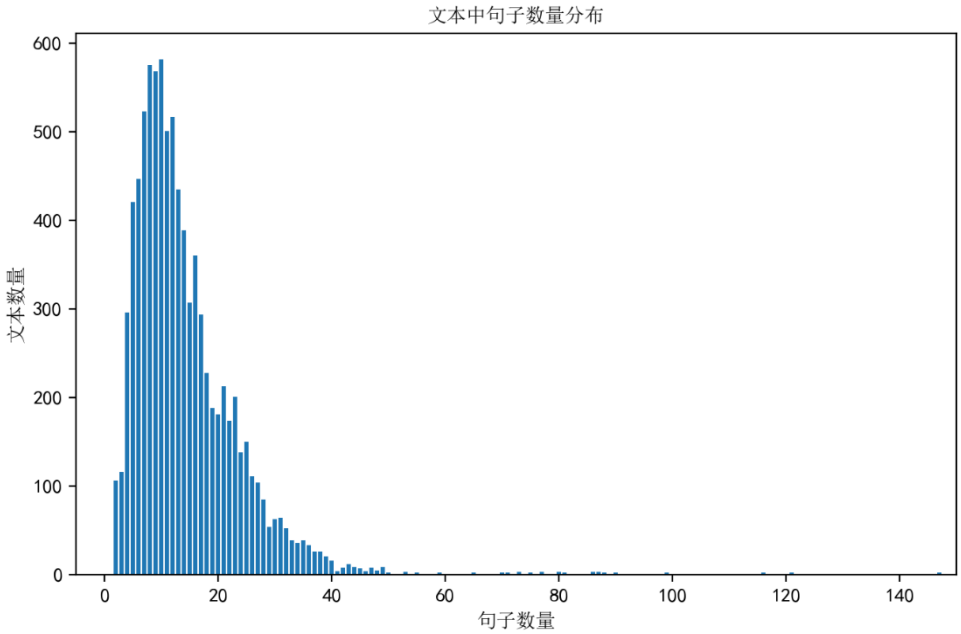

图 5-4 文本子句数量分布
Fig 5-4 Number Distribution of Text Sentences

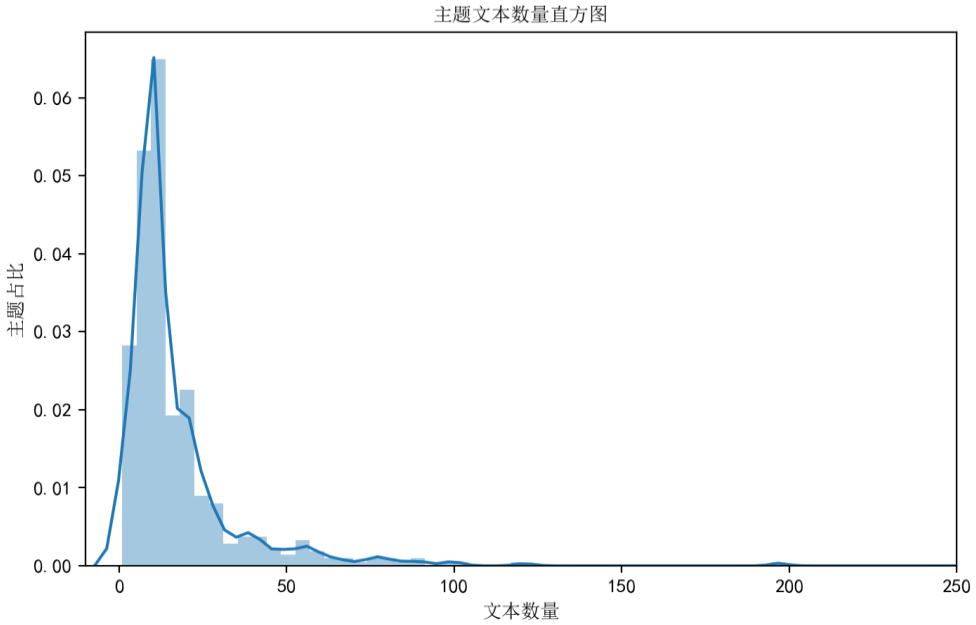

图 5-5 主题文本数量直方图
Fig 5-5 Topic Text Quantity Histogram

5.3.2 词性标注

现有的词性标注开源工具有很多，一般都集成在分词工具中，包括盘古分词、Yaha 分词、Jieba 分词、清华 THULAC 等。在这里，本文选择使用 Jieba 分词工具。

本文将专利文本输入到 Jieba 分词工具中进行词性标注和预分词。在后面步骤中将已经完成词性标注的各个单词，结合专利文本中合成型技术专有名词的构词模式识别出专利文本中合成型技术专有名词，然后再将其扩展入自定义词典中。

5.3.3 合成型技术专有名词的识别

识别专利文本中的合成型技术专有名词对于度量专利文本相似度至关重要。在专利文本中包含了众多技术专有名词，这些技术专有名词构成了该专利各方面的属性。这些技术专有名词可以在很大程度上表述该专利的技术特征。换句话说，如果成功识别出了专利文本中的技术专有名词，那么就能够用它表征专利的很大一部分信息。然而这些技术专有名词又有很大一部分是合成型技术专有名词，直接使用分词算法难以进行准确的识别，容易被误切分成多个词。

根据第三章所提出的合成型技术专有名词的构词模式，利用模式与字符串相匹配的方式识别出专利数据集中的合成型技术专有名词。在专利数据集中运行合成型技术专有名词算法后，对 35993 篇专利文本提取出来了 125754 个合成型技术专有名词。

5.3.4 中文分词

正如前文所述，现有的分词工具有很多。在这里选择使用 Jieba 分词工具，主要是考虑到 Jieba 分词有以下三点优势：（1）已经很好的嵌入到了 Python 库；（2）支持扩充自定义词典，若待分词文本中存在自定义词典中的单词，将使用机械分词进行切分，在机械分词的基础上再使用统计学分词的方法，可以有效利用先验知识对那些统计机器学习分词算法无法正确切分的单词进行准确的切分，有效提高分词准确率；（3）能够实现词性标注。基于以上三点优势，本文选择使用的是 Jieba 分词。

在完成前文所述的合成型技术专有名词识别，并将识别出来的合成型技术专有名词加入到 Jieba 分词的自定义词典后，能够极大的提高分词算法对于专有名词的识别率。

5.3.5 停用词处理

本文的目标是度量专利文本之间的相似性，专利文本中的技术专有名词和谓语动词可以作为关键词。对于不影响专利文本中技术表述的单词可以视为停用词进行过滤。停用词大多分为连接词、介词、代词、标点符号、语气词。我们利用各个研究机构开源的停用词表，再根据语料与需求自行扩充得到专利领域的停用词表。事实上，某些动词和名词应用十分广泛，但是这些词不能够表征文本的关键信息，难以帮助缩小相似专利匹配范围，同时还会提高运算复杂度，所以通常会在文本中将这些词去掉，从而提高匹配性能。

本文选择过滤的停用词类型及部分停用词如表 2 所示。通过字符串匹配的方式，删除专利文本中出现在停用词表中的单词以达到去除停用词的目的。我们所构建的停用词表包含 13882 个停用词。

表 2 部分停用词
Table 2 Partial Discontinuation

类型	部分停用词
连接词	和、并且、不仅、因此、不过、但是、而且
介词	在于、与、的、为、归于
标点符号	“，”、“、”、“！”、“？”、“。”
副词	非常、及其、绝对、十分、最、更
代词	我们、我、其、它、那个
量词	一台、一段、一种、第一种、步骤一、一侧
部分名词	趋势、简称、元件、组件、设备

5.3.6 关键词提取

在自然语言处理领域，对于文本本身而言最关键的是要把最重要的信息提取出来。而无论是对于长文本还是短文本，往往可以通过几个关键词窥探整个文本的主题思想。关键词提取的准确程度直接关系到文本相似性度量的最终效果。同时，如果两篇文本中的关键词越相似，则它们的内容也就越相似。所以能够高效准确的提取出专利文本中的关键词对于度量专利文本相似性至关重要。

对于合成型技术专有名词而言它们的词性为名词，且它一定是专利文本中的关键词。由 3.1.2 小节所述，对于专利文本中的少量关键词可以被正确切分，在合成型技术专有名词提取部分没有将其提取出来。由于在分词阶段已经对文本中各

个单词实现了词性标注，同时考虑到专利文本中各个“子句”结构大多为 SAO 结构、AO 结构、SA 结构，因此能够表征专利文本重要信息的单词的词性大多为名词词性或动词词性，所以在去完停用词后对剩下单词提取词性为名词和动词的单词作为专利文本中的关键词。

专利测试集以及专利数据库中专利文本的关键词数量在各个文档中的分布情况如图 5-6 所示，该图的横坐标表示文本中关键词数量 x，纵坐标表示文本数 y，该图表示有 y 个文本包含 x 个关键词。由此图可知大多数文本的关键词个数都在 10-50 之间，数据分布较为均衡。

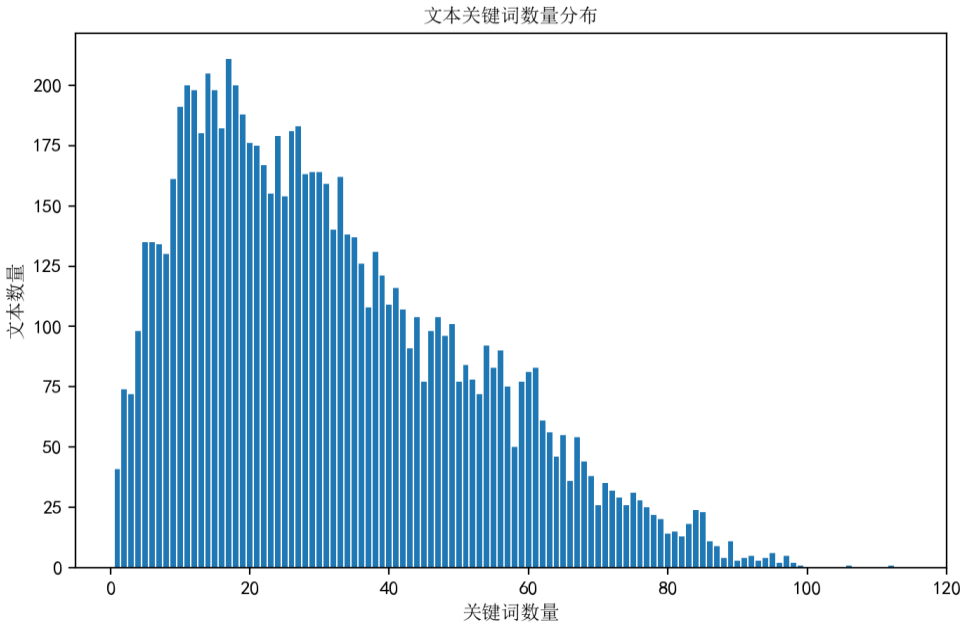

图 5-6 文本关键词数量分布
Fig 5-6 Number Distribution of Text Keyword

5.4 专利文本向量化

为了更好的度量单词之间的语义相似性，本文使用 Word2vec 模型对专利主权项提取的关键词进行词向量训练。在训练词向量过程中，Word2vec 模型还有许多超参数需要设置，例如滑动窗的大小、词向量的维度等。

值得注意的是，由于预先并不知道单词之间的相似性，所以不能直接设定模型的超参数。为了解决这一问题，本文提出了一种基于同义词相似度分析的方法来间接的评估 Word2vec 算法的向量化效果。首先本文会从文本中提取出一系列同义词，例如：调控、调节；水下声纳、水下声呐；检测、测试；采集、采样等。

通过不断调整模型参数，计算这些同义词的平均相似度，理想情况下，这些同义词通过 Word2vec 模型向量化表征后，词向量的相似度越高越好，因此最终取这些同义词整体平均相似度最高的情况下的模型超参数作为最终的超参数。

通过不断的调整超参数，在以下超参数下上述同义词的平均相似度最高：词向量维度为 350 维、高频词汇随机采样的配置阈值为 2e-3、使用负采样算法、负采样的个数为 5、使用 Skip-Gram 模型进行训练、窗口大小设置为 5、需要计算词向量的最小词频为 1、迭代次数为 15、在随机梯度下降法中迭代的初始步长设置为 0.005、最小的迭代步长值设置为 0.00005。因此认为在此超参数下，词向量的训练效果最好。

5.5 专利文本相似度算法实现及结果分析

将预处理好的文本利用本文设计的基于句法表征的专利文本相似度算法计算专利文本之间的相似度，先利用“子句”中单词的句法成分和语义相似性计算专利文本各个“子句”之间的相似度，然后再通过专利文本各个“子句”之间的相似度利用 DTW 算法计算文本相似度。最后为每个待查找专利从专利库中搜索出五个相似专利。根据知网已经给出各个专利的相似专利作为依据评估算法的准确率。除此以外本文还在同一数据集下使用一些基于文本语义信息计算文本相似度的算法作为对比实验，包括：（1）Doc2vec；（2）LDA；（3）VSM；（4）WMD 算法。对比实验结果如图 5-7 所示。

相对于 WMD 算法准确率提高了 1.6%，主要原因在于本算法考虑到了专利文本的语法结构表征，在度量专利文本相似性时，更加充分的利用了文本信息。

本算法相对于传统文本相似度算法而言都有较大的提升，相对于 doc2vec 算法，准确率提高了 14%；相对于 LDA 算法，准确率提高了 15.2%；相对于 VSM 算法，准确率提高了 2.8%。本文认为这是由于，在度量文本相似性时，将文本切割成更小的单元进行相似性分析，可以更加充分利用专利文本的语义信息。

在几个传统算法中，VSM 的效果甚至要好于 Doc2vec 和 LDA。由于 VSM 是以 Onehot 的形式生成文本向量，所以在比较专利文本相似性时，VSM 模型本质上是在比较在两篇文本中共现的词的数量。而在比较专利文本相似性中，专利中的关键词极其特殊，只有在一类特定的专利文本中才会出现该关键词，所以通过 VSM 模型来计算两篇文本共现的关键词的数量，共现的关键词越多，则这两篇文本讲述相似技术的概率越大。而对于 Doc2vec 而言，它虽然是利用文本的语言特性将专利文本转化成了一个向量化表示，但是由于它的原理和 Word2vec 相似，区别只是在于在训练句向量的过程中，Doc2vec 加入了文本中的词序特征，但是本质上还

是在对专利文本中的各个词向量求和取平均得到句向量，因此得到的文本句向量湮没掉了文本中关键词的词向量的特殊性。对于 LDA 而言，它是将专利文本映射到 *K* 个主题，生成主题概率分布。通过比较文本-主题向量来比较专利文本之间的相似性。但是不能认为主题相似的专利之间表述的专利技术相似。从对比实验中也在一定程度上验证了本文提出的算法的合理性，即：比较专利文本相似最重要的是比较专利文本关键词的相似度，这也在一定程度上解释了 VSM 算法的效果要优于 Doc2vec 和 LDA 的原因。

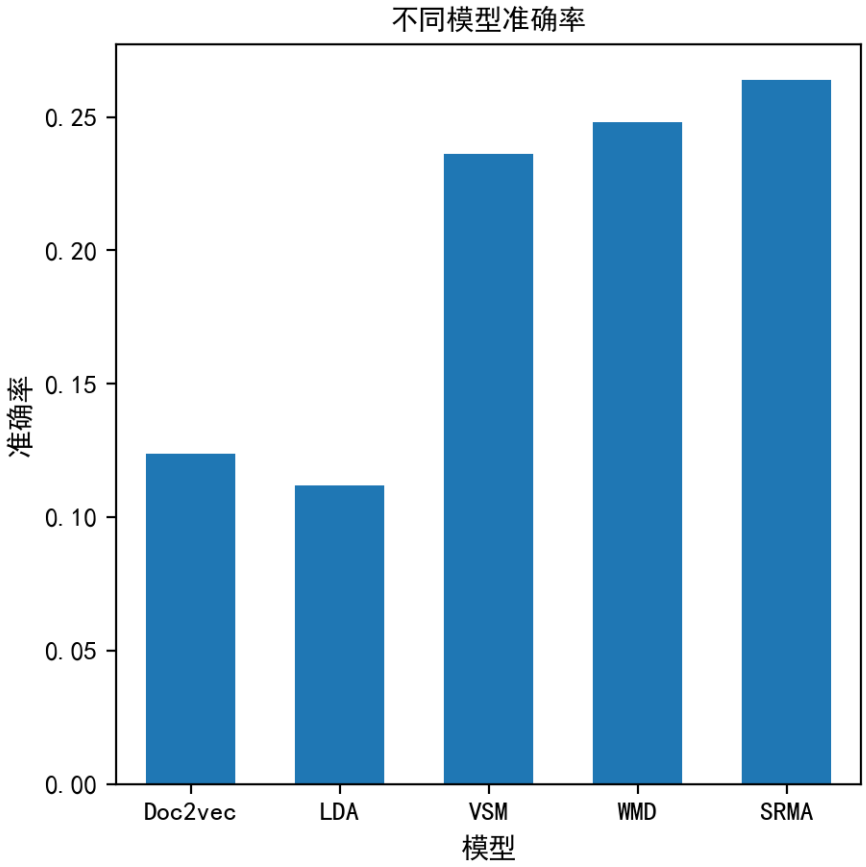

图 5-7 实验结果对比
Fig 5-7 Comparison of experimental result

对于 Doc2vec 而言，它虽然利用文本的语言特性将专利文本转化成了向量化表示，但是由于它的原理和 Word2vec 相似，区别只是在于在训练句向量的过程中，Doc2vec 加入了文本中的词序特征，但是本质上还是在对专利文本中的各个词向量求和取平均得到句向量，因此得到的文本句向量湮没掉了文本中关键词的特殊性。对于 LDA 而言，它是将专利文本映射到 *K* 个主题生成一个概率分布，通过比较文

本-主题向量来比较专利文本之间的相似性。但是不能认为主题相似的专利表述的专利技术也相似。

在本实验中，无论是专利测试数据还是专利数据库，它们都同属于电子领域，对于这些专利文本而言，无论它是否是相似专利，它们之间本身就存在很多相似关键词，所以这也在一定程度上影响了最终的实验结果。

本论文所研究方法是基于无监督学习的方式实现的，与基于有监督的分类问题、回归问题大不相同。因此自然不能和通过有监督学习问题的结果相提并论。同时对于一篇专利而言，在专利库中仅存在少量几篇专利和它是相似专利，而其它的所有专利都和它不相似，在本实验的测试集中，平均对于每一个测试专利，专利数据库中有超过 99.8%的专利与它不相似，在海量专利中找出少量几篇相似专利，难度也非常大。

5.6 本章小结

本章介绍了算法整体，并且分步骤的阐述了实现方式。首先对专利数据爬虫进行了简要介绍，并对专利数据集数据特征分布进行了简要的分析；然后简要介绍了数据预处理过程；接着对 Word2vec 词向量训练调参过程进行了简单阐述；最后通过将 SRMA 算法与现有的文本相似度算法在同一数据集上进行对比实验，证明了本论文所提出算法的有效性。

6 结论

在自然语言处理领域，文本相似性问题一直是一个热点问题，也是一个难点问题。近些年来虽然很多学者提出了文本相似性算法并应用于专利领域，但大多不能从语义层面很好的度量专利文本相似性。数据挖掘、机器学习和深度学习技术在自然语言处理各个领域得到了广泛发展和应用，同时也为解决专利文本相似性问题提供了新思路。

6.1 本文主要工作

本文的主要工作分为两部分：第一，文本预处理，其中主要贡献在于提出一种利用构词规则的合成型技术专有名词识别技术；第二，专利文本相似性度量，其中主要贡献在于利用文本结构和语义信息来度量文本相似性。本小节主要对这两方面内容按照其实现步骤进行叙述。

第一步，提取专利文本中的关键词，并针对专利文本中出现的合成型技术专有名词根据其构词特点进行规则模式定义，通过模式和字符串相匹配的方式高效准确的识别出专利文本中的合成型技术专有名词，提高关键词的提取准确率；

第二步，利用 Word2vec 算法计算文本中关键词的语义相似性；

第三步，为了利用专利文本“子句”中的 SAO 结构、SA 结构、AO 结构，将专利主权项文本以逗号、分号、句号为单位切割成“子句”集合，对专利文本各个“子句”内部中的关键词进行位置编码，通过关键词之间的位置编码值来区分关键词的重要性程度；

第四步，综合专利“子句”间各个关键词的语义相似性和重要性，基于具有偏好序的稳定匹配算法计算专利文本各个“子句”之间的相似度，并根据“子句”之间的相似度利用 DTW 算法计算专利文本之间的相似度；

第五步，通过实验验证了本研究的有效性和合理性。实验结果表明本文提出的针对专利文本句式结构所制定的专利文本相似度算法相对于传统算法效果更好。

6.2 未来工作展望

第一，关于专利文本中关键词的识别部分可以做到进一步的优化。首先，可以总结出更丰富的构词模式，利用模式匹配的方式识别出专利文本中更多的合成型技术专有名词；其次，还可以通过统计机器学习的方法，基于大量的标注语料

进行专利文本关键词识别的学习，进而实现专利文本中的关键词识别。

第二，能够准确的识别专利文本的基本句式对于专利文本相似性的度量至关重要。在识别专利文本的基本句式时，可以不仅仅依靠词性判断。随着自然语言处理技术的发展，使得更为高效且准确的识别专利文本的句式结构成为可能。

第三，专利库中专利数量越多，从专利库中准确搜索出专利文本的相似专利的难度越大，这就对专利相似度算法提出了更高的要求。本文认为，还可以在进行专利相似度计算之前进行召回处理，从专利库中排除掉明显和该专利不相似的专利，在提高专利相似度算法准确率的基础上还可以降低运算量。

参考文献

[1] Shin J, Yongtae Park. Generation and Application of Patent Claim Map: Text Mining and Network Analysis [J]. Journal of Intellectual Property Rights, 2005, 10(3):198-205.

[2] Lerner J. The Importance of Patent Scope: An Empirical Analysis[J]. The RAND Journal of Economics, 1994, 25(2):319-333.

[3] Ernst H. Patent applications and subsequent changes of performance: evidence from time-series cross-section analyses on the firm level[J]. Research Policy, 2001, 30(1): 143-157.

[4] Shane S. Technological opportunities and new firm creation[J]. Management science, 2001, 47(2): 205-220.

[5] Takeuchi H, Nonaka I. The new new product development game[J]. Harvard business review, 1986, 64(1): 137-146.

[6] Users L. A Source of Novel Product Concepts[J]. Mana, 1986.

[7] Campbell R S. Patent trends as a technological forecasting tool[J]. World Patent Information, 1983, 5(3): 137-143.

[8] 张海超. 基于 SAO 的中文相似专利识别方法及其应用研究[D]. 北京工业大学, 2015.

[9] 杨春明, 韩永国. 快速的领域文档关键词自动提取算法[J]. 计算机工程与设计, 2011, 32(6):2142-2145.

[10] Fujii A, Iwayama M, Kando N. Introduction to the special issue on patent processing[J]. Information Processing & Management, 2007, 43(5): 1149-1153.

[11] Tseng Y H. Text mining for patent map analysis[J]. Catalyst, 2005, 5424054(5780101): 6333016.

[12] Tseng Y H, Lin C J, Lin Y I. Text mining techniques for patent analysis[J]. Information Processing & Management, 2007, 43(5): 1216-1247.

[13] Tseng Y H, Wang Y M, Lin Y I, et al. Patent surrogate extraction and evaluation in the context of patent mapping[J]. Journal of Information Science, 2007, 33(6): 718-736.

[14] Jeong C, Kim K. Creating patents on the new technology using analogy-based patent mining[J]. Expert Systems with Applications, 2014, 41(8): 3605-3614.

[15] Grawe M F, Martins C A, Bonfante A G. Automated patent classification using word embedding[C]//2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2017: 408-411.

[16] Ashtekar A, Singh P. Loop quantum cosmology: a status report[J]. Classical and Quantum Gravity, 2011, 28(21): 213001.

[17] Grawe M F, Martins C A, Bonfante A G. Automated Patent Classification Using Word Embedding[C]// IEEE International Conference on Machine Learning & Applications. IEEE, 2018.

[18] Linsey J S. Design-by-analogy and representation in innovative engineering concept generation[D]. , 2007.

[19] McAdams D A, Wood K L. A quantitative similarity metric for design-by-analogy[J]. Transactions-American Society of Mechanical Engineers Journal of Mechanical Design, 2002, 124(2): 173-182.

[20] Verhaegen P A, D'hondt J, Vandevenne D, et al. Identifying candidates for design-by-analogy[J]. Computers in Industry, 2011, 62(4): 446-459.

[21] Bhatta S R, Goel A K. From design experiences to generic mechanisms: model-based learning in analogical design[J] . AI EDAM, 1996, 10(2): 131-136.

[22] Wang H, Ohsawa Y. Idea discovery: A scenario-based systematic approach for decision making in market innovation [J]. Expert Systems with Applications, 2013, 40(2):429-438.

[23] Tiwana S, Horowitz E. Extracting problem solved concepts from patent documents[C]// International Workshop on Patent Information Retrieval. ACM, 2009.

[24] Smith H. Automation of patent classification[J]. World Patent Information, 2002, 24(4): 269-271.

[25] Arts S, Cassiman B, Gomez J C. Text matching to measure patent similarity[J]. Working Papers Department of Managerial Economics, Strategy and Innovation (MSI), 2017, 39(1).

[26] Lee C, Song B, Park Y. How to assess patent infringement risks: a semantic patent claim analysis using dependency relationships[J].Technology Analysis and Strategic Management, 2013, 25(1).

[27] Wang C, Song Y, Li H, et al. KnowSim: A Document Similarity Measure on Structured Heterogeneous Information Networks[J]. 2015.

[28] Sharma P, Tripathi R, Singh V K, et al. Automated patents search through semantic similarity[C]// International Conference on Computer. IEEE, 2016.

[29] Wang W M, Cheung C F. A Semantic-based Intellectual Property Management System (SIPMS) for supporting patent analysis[J]. Engineering Applications of Artificial Intelligence, 2011, 24(8):1510-1520.

[30] Sharma P, Tripathi R, Tripathi R C. Finding similar patents through semantic expansion[C]// International Conference on Computer Communication & Informatics. IEEE, 2016.

[31] Uchida H, Mano A, Yukawa T. Patent Map Generation Using Concept-Based Vector Space Model[C]//NTCIR. 2004.

[32] 张玉洁. 英汉专利翻译技巧浅析[J]. 新丝路(下旬), 2016(3):121-121.

[33] 李熙, 徐德智. 基于 WordNet 的概念语义相似度研究[J]. 湖南科技学院学报, 2008, 29(12):115-116.

[34] Dong Z, Dong Q, Ebrary I. HowNet and the computation of meaning[J]. 2015.

[35] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine Learning research, 2003, 3(Jan): 993-1022.

[36] 居斌. 潜在语义标引在中文信息检索中的研究与实现[J]. 计算机工程, 2007, 33(5):193-196.

[37] 孙海霞, 钱庆, 成颖. 基于本体的语义相似度计算方法研究综述[J]. 数据分析与知识发现, 2001, 26(1): 51-56.

[38] Jiang J, Conrath D. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy[C]. Proceedings of International Conference Research on Computational Linguistics,1997:19–33.

[39] Mikolov, Tomas, Chen, Kai, Corrado, Greg, et al. Efficient estimation of word representations in vector space[J]. Computer Science, 2013.

[40] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]. International Conference on Neural Information Processing Systems. Curran Associates Inc. 2013:3111-3119.

[41] Le Q, Mikolov T. Distributed representations of sentences and documents[C]//International conference on machine learning. 2014: 1188-1196.

[42] 金博, 史彦军, 滕弘飞. 基于语义理解的文本相似度算法[J]. 大连理工大学学报, 2005, 45(2).

[43] 徐德智. 基于本体的概念间语义相似度计算方法研究[J]. journal6, 2006, 43(8):154-156.

[44] Chi T, Wang H, Liu L, et al. Text similarity calculation method based on ontology model[C]//Cloud Computing and Internet of Things (CCIOT), 2014 International Conference on. IEEE, 2014.

[45] Rahutomo F, Aritsugi M. Econo-ESA in semantic text similarity[J]. Springerplus, 2014, 3:149(3).

[46] 曹卫峰. 中文分词关键技术研究[D]. 南京理工大学,2009.

[47] 王靖. 基于机械切分和标注的中文分词研究[D]. 湖南大学,2009.

[48] 苏勇. 基于理解的汉语分词系统的设计与实现[D]. 电子科技大学,2011.

[49] 兰冲. 基于统计规则的中文分词研究[D]. 西安电子科技大学,2011.

[50] 易剑波. 基于文本挖掘的电商用户评论分析与系统实现[D]. 2017.

[51] 杨荣根, 杨忠. 基于 HMM 中文词性标注研究[J]. 金陵科技学院学报, 2017(1).

[52] Jurafsky D, Martin J H. Part of Speech Tagging[J]. Speech and language processing, 2016.

[53] 张卫. 中文词性标注的研究与实现[D]. 南京师范大学, 2007.

[54] Schofield A, Magnusson M, Mimno D. Pulling Out the Stops: Rethinking Stopword Removal for Topic Models[C]// Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. 2017:432-436.

[55] Hao L, Hao L. Automatic Identification of Stop Words in Chinese Text Classification[C]// International Conference on Computer Science and Software Engineering. IEEE, 2008:718-722.

[56] Mnih A, Hinton G. A scalable hierarchical distributed language model[C]. International Conference on Neural Information Processing Systems. Curran Associates Inc. 2008:1081-1088.

[57] Rong X. word2vec Parameter learning explained[J]. Computer Science, 2014.

[58] 郭庆琳, 李艳梅, 唐琦. 基于 VSM 的文本相似度计算的研究[J]. 计算机应用研究, 2008, 25(11):3256-3258.

[59] 吴多坚. 基于 word2vec 的中文文本相似度研究与实现[D]. 2016.

[60] Kusner M J, Sun Y, Kolkin N I, et al. From word embeddings to document distances[C]// International Conference on International Conference on Machine Learning. JMLR.org, 2015.

[61] 李巍, 郭强, 曹华. 具有成功率约束的最优匹配问题[J]. 计算机工程与应用, 2011, 47(4): 33-35.

[62] 主谓宾结构. https://www.zybang.com/question/5b00c5fe3a5ec9e1095ed98abc635c2c.html

[63] 主谓结构. https://baike.baidu.com/item/主谓结构/8982620?fr=aladdin.html

[64] 动宾结构. https://baike.baidu.com/item/动宾结构/1879958.html

[65] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.

[66] Liu G. The semantic vector space model (SVSM): a text representation and searching

technique[C]// Twenty-seventh Hawaii International Conference on System Sciences. 1994.

附录

中文词性表

Ag	形语素	形容词性语素。形容词代码为 a，语素代码 g 前面置以 A。
a	形容词	取英语形容词 adjective 的第 1 个字母。
ad	副形词	直接作状语的形容词。形容词代码 a 和副词代码 d 并在一起。
an	名形词	具有名词功能的形容词。
b	区别词	取汉字“别”的声母。
c	连词	取英语连词 conjunction 的第 1 个字母。
dg	副语素	副词性语素。副词代码为 d，语素代码 g 前面置以 D。
d	副词	取 adverb 的第 2 个字母，因其第 1 个字母已用于形容词。
e	叹词	取英语叹词 exclamation 的第 1 个字母。
f	方位词	取汉字“方”。
g	语素	绝大多数语素都能作为合成词的“词根”。
h	前接成分	取英语 head 的第 1 个字母。
i	成语	取英语成语 idiom 的第 1 个字母。
j	简称略语	取汉字“简”的声母。
k	后接成分	
l	习用语	习用语尚未成为成语，有点“临时性”，取“临”的声母。
m	数词	取英语 numeral 的第 3 个字母，n，u 已有他用。
Ng	名语素	名词性语素。名词代码为 n，语素代码 g 前面置以 N。
n	名词	取英语名词 noun 的第 1 个字母。
nr	人名	名词代码 n 和“人（ren）”的声母并在一起。
ns	地名	名词代码 n 和处所词代码 s 并在一起。
nt	机构团体	“团”的声母为 t，名词代码 n 和 t 并在一起。
nz	其他专名	“专”的声母的第 1 个字母为 z，名词代码 n 和 z 并在一起。
o	拟声词	取英语拟声词 onomatopoeia 的第 1 个字母。
p	介词	取英语介词 prepositional 的第 1 个字母。
q	量词	取英语 quantity 的第 1 个字母。
r	代词	取英语代词 pronoun 的第 2 个字母,因 p 已用于介词。
s	处所词	取英语 space 的第 1 个字母。

中文词性表（分表）

vg	动语素	动词性语素。动词代码为 v。在语素的代码 g 前面置以 V。
tg	时语素	时间词性语素。时间词代码为 t,在语素的代码 g 前面置以 T。
t	时间词	取英语 time 的第 1 个字母。
u	助词	取英语助词 auxiliary。
v	动词	取英语动词 verb 的第一个字母。
vd	副动词	直接作状语的动词。动词和副词的代码并在一起。
vn	名动词	指具有名词功能的动词。动词和名词的代码并在一起。
w	标点符号	
x	非语素字	非语素字只是一个符号，字母 x 通常用于代表未知数、符号。
y	语气词	取汉字“语”的声母。
z	状态词	取汉字“状”的声母的前一个字母。
un	未知词	不可识别词及用户自定义词组。取英文 Unkonwn 首两个字母。

作者简历及攻读硕士学位期间取得的研究成果

陈泽龙，男，1995 年 8 月生。2013 年 9 月至 2017 年 6 月就读于石家庄铁道大学通信工程专业，取得工学学士学位。2017 年 9 月至 2019 年 6 月就读于北京交通大学电子与通信工程专业，研究方向是信息网络，取得工学专业硕士学位。攻读专业硕士学位期间，主要从事专利文本相似性度量方面的研究工作。

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：陈泽龙 签字日期：2019 年 6 月 3 日

学位论文数据集

表 1.1： 数据集页

关键词*	密级*	中图分类号	UDC	论文资助
专利侵权；专利相似度；语义信息；句式结构；时间序列；DTW 算法	公开			
学位授予单位名称*		学位授予单位代码*	学位类别*	学位级别*
北京交通大学		10004	专业学位	硕士
论文题名*		并列题名		论文语种*
基于句法表征的专利文本相似性评估				中文
作者姓名*	陈泽龙		学号*	17125009
培养单位名称*		培养单位代码*	培养单位地址	邮编
北京交通大学		10004	北京市海淀区西直门外上园村 3 号	100044
学科专业*		研究方向*	学制*	学位授予年*
电子与通信工程		信息网络	2	2019
论文提交日期*	2019.6.3			
导师姓名*	郑宏云		职称*	副教授
评阅人	答辩委员会主席*		答辩委员会成员	
	沈波		王目光、赵永祥、陶丹、徐少毅	
电子版论文提交格式　文本（ ）　图像（ ） 视频（ ） 音频（ ） 多媒体（ ） 其他（ ） 推荐格式：application/msword；application/pdf				
电子版论文出版（发布）者		电子版论文出版（发布）地		权限声明
论文总页数*	58			
共 33 项，其中带*为必填数据，为 21 项。				