Spring 5-1-2020

Table-to-Text: Generating Descriptive Text for Scientific Tables from Randomized Controlled Trials

Qiang Wei

Table-to-Text: Generating Descriptive Text for Scientific Tables from Randomized Controlled Trials

A
Dissertation

Presented to the Faculty of
The University of Texas
Health Science Center at Houston
School of Biomedical Informatics
in Partial Fulfilment of the Requirements for the Degree of

Doctor of Philosophy

By

Qiang Wei, M.S.

University of Texas Health Science Center at Houston

2020

Dissertation Committee:

Hua Xu, PhD[1], Advisor
Cui Tao, PhD[1]
Qiaozhu Mei, PhD[2]

[1]The School of Biomedical Informatics
[2]University of Michigan, School of Information

Acknowledgements

First, I would like to express my special appreciation to my advisor Dr. Hua Xu for his continuous guidance and support during my PhD study. Without his guidance and encouragement this PhD would not have been possible.

I would also like to express my gratitude to my thesis committee, Dr. Cui Tao and Dr. Qiaozhu Mei, for their comments and suggestions, for reviewing the manuscripts, and during the research for this thesis project. Many thanks to Dr. Cui Tao for her guidance on the establishment of the information model, and Dr. Qiaozhu Mei for the development of the deep learning methods for text generation.

I am grateful to all of those with whom I have had the pleasure to work during this and other related projects, including members and alumni of the Xu's Lab and all my friends in SBMI. Thanks to Jingqi Wang and Dr. Jingcheng Du for their help in development of the methods for recognizing tables and text generation. Thanks to Yujia Zhou, Bo Zhao, Xiao Dong, Jin Ding, Xinyue Hu, and Zaid Soomro for their help in development of the annotation guideline and annotating the dataset.

Lastly, I wish to thank my family: my parents and my girlfriend Kun Nie, for supporting me emotionally throughout my PhD study and the completion of this thesis project.

ABSTRACT

Unprecedented amounts of data have been generated in the biomedical domain, and the bottleneck for biomedical research has shifted from data generation to data management, interpretation, and communication. Therefore, it is highly desirable to develop systems to assist in text generation from biomedical data, which will greatly improve the dissemination of scientific findings. However, very few studies have investigated issues of data-to-text generation in the biomedical domain. Here I present a systematic study for generating descriptive text from tables in randomized clinical trials (RCT) articles, which includes: (1) an information model for representing RCT tables; (2) annotated corpora containing pairs of RCT table and descriptive text, and labeled structural and semantic information of RCT tables; (3) methods for recognizing structural and semantic information of RCT tables; (4) methods for generating text from RCT tables, evaluated by a user study on three aspects: relevance, grammatical quality, and matching. The proposed hybrid text generation method achieved a low bilingual evaluation understudy (BLEU) score of 5.69; but human review achieved scores of 9.3, 9.9 and 9.3 for relevance, grammatical quality and matching, respectively, which are comparable to review of original human-written text. To the best of our knowledge, this is the first study to generate text from scientific tables in the biomedical domain. The proposed information model, labeled corpora and developed methods for recognizing tables and generating descriptive text could also facilitate other biomedical and informatics research and applications.

Vita

2009 ……. Bachelor of Science, Bioinformatics, Zhejiang University

2013 ……... Master of Science, Bioinformatics, Zhejiang University

2014 to present ……. School of Biomedical Informatics, The University of Texas Health Science Center at Houston

Publications

- Wu Stephen, Roberts Kirk, Datta Surabhi, Du Jingcheng, Ji Zongcheng, Si Yuqi, Soni Sarvesh, Wang Qiong, **Wei Qiang**, Xiang Yang, Zhao Bo, Xu Hua. Deep learning in clinical natural language processing: a methodical review. Journal of the American Medical Informatics Association : JAMIA 2020;27:457–70.
- **Wei Qiang**, Ji Zongcheng, Li Zhiheng, Du Jingcheng, Wang Jingqi, Xu Jun, Xiang Yang, Tiryaki Firat, Wu Stephen, Zhang Yaoyun, Tao Cui, Xu Hua. A study of deep learning approaches for medication and adverse drug event extraction from clinical text. Journal of the American Medical Informatics Association : JAMIA 2020;27:13–21.
- Xu Jun, Li Zhiheng, **Wei Qiang**, Wu Yonghui, Xiang Yang, Lee Hee Jin, Zhang Yaoyun, Wu Stephen, Xu Hua. Applying a deep learning-based sequence labeling approach to detect attributes of medical concepts in clinical text. BMC Medical Informatics and Decision Making 2019;19:236.
- **Wei Qiang**, Ji Zongcheng, Si Yuqi, Du Jingcheng, Wang Jingqi, Tiryaki Firat, Wu Stephen, Tao Cui, Roberts Kirk, Xu Hua. Relation Extraction from Clinical Narratives Using Pre-trained Language Models. In: AMIA Annual Symposium Proceedings. 2019.
- **Wei Qiang**, Chen Yukun, Salimi Mandana, Denny Joshua C, Mei Qiaozhu, Lasko Thomas A, Chen Qingxia, Wu Stephen, Franklin Amy, Cohen Trevor, Xu Hua. Cost-aware active

learning for named entity recognition in clinical text. Journal of the American Medical Informatics Association : JAMIA 2019;26:1314–22.

- **Wei Qiang**, Zhang Yaoyun, Amith Muhammad, Lin Rebecca, Lapeyrolerie Jenay, Tao Cui, Xu Hua. Recognizing software names in biomedical literature using machine learning. Health Informatics Journal Published Online First: 2019.
- Ji Zongcheng, **Wei Qiang**, Xu Hua. BERT-based Ranking for Biomedical Entity Normalization. Published Online First: 9 August 2019.
- Du Jingcheng, Luo Chongliang, **Wei Qiang**, Chen Yong, Tao Cui. Exploring difference in public perceptions on HPV vaccine between gender groups from Twitter using deep learning. Published Online First: 6 July 2019.
- Xu Jun, Li Zhiheng, **Wei Qiang**, Wu Yonghui, Xiang Yang, Lee Hee Jin, Zhang Yaoyun, Wu Stephen, Xu Hua. Detect attributes of medical concepts via sequence labeling. In: 2019 IEEE International Conference on Healthcare Informatics, ICHI 2019. Institute of Electrical and Electronics Engineers Inc. 2019. 1–2.
- Ji Zongcheng, **Wei Qiang**, Franklin Amy, Cohen Trevor, Xu Hua. Cost-sensitive Active Learning for Phenotyping of Electronic Health Records. In: AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science. American Medical Informatics Association 2019. 829–38.
- Du Jingcheng, Zhang Yaoyun, Luo Jianhong, Jia Yuxi, **Wei Qiang**, Tao Cui, Xu Hua. Extracting psychiatric stressors for suicide from social media using deep learning. BMC Medical Informatics and Decision Making 2018;18.
- Tang Lingyi, Xu Jun, Hu Xinyue, **Wei Qiang**, Xu Hua. Building a Biomedical Chinese-English Parallel Corpus from MEDLINE. In: MultilingualBIO: Multilingual Biomedical Text Processing. 2018. 8.
- Zhang Yaoyun, **Wei Qiang**, Wang Jingqi, Xu Hua. CLAMP-PA: A machine learning based pre-annotation pipeline for corpus construction of clinical concepts. In: AMIA Annual Symposium Proceedings,. 2018.
- **Wei Qiang**, Franklin Amy, Cohen Trevor, Xu Hua. Clinical text annotation - what factors are associated with the cost of time? AMIA . Annual Symposium proceedings AMIA Symposium 2018;2018:1552–60.

- Xu Jun, Lee Hee-Jin, Ji Zongcheng, Wang Jingqi, **Wei Qiang**, Xu Hua. UTH_CCB System for Adverse Drug Reaction Extraction from Drug Labels at TAC-ADR 2017. In: TAC. 2017.
- Sun Jingchun, **Wei Qiang**, Zhou Yubo, Wang Jingqi, Liu Qi, Xu Hua. A systematic analysis of FDA-approved anticancer drugs. BMC Systems Biology 2017;11.
- Soysal E, Lee HJ, Zhang Y, Huang LCC, Chen X, Wei Q, Zheng W, Chang JT, Cohen T, Sun J, Xu H. CATTLE (CAncer treatment treasury with linked evidence): An integrated knowledge base for personalized oncology research and practice. CPT: Pharmacometrics and Systems Pharmacology 2017;6:188–96.
- **Wei Qiang**, Chen Yukun, Moon Sungrim, Cohen Trevor, Xu Hua. A Study of Active Learning for Document Selection in Clinical Named Entity Recognition. AMIA 2016.
- Xu Jun, Wu Yonghui, Zhang Yaoyun, Wang Jingqi, Liu Ruiling, **Wei Qiang**, Xu Hua. UTH-CCB@BioCreative V CDR Task: Identifying Chemical-induced Disease Relations in Biomedical Text. In: Proceedings of the Fifth BioCreative Challenge Evaluation Workshop. 2015. 254–9.
- Liu Yanbin, **Wei Qiang**, Yu Guisheng, Gai Wanxia, Li Yongquan, Chen Xin. DCDB 2.0: a major update of the drug combination database. Database : the journal of biological databases and curation 2014;2014:bau124.
- Zhou Xi, Chen Pengcheng, **Wei Qiang**, Shen Xueling, Chen Xin. Human interactome resource and gene set linkage analysis for the functional interpretation of biologically meaningful gene sets. Bioinformatics (Oxford, England) 2013;29:2024–31.

Table of Contents

Acknowledgements i

ABSTRACT ii

Vita iii

List of Tables viii

List of Figures ix

I. Introduction 1

2. Background 3
2.1 Relevant work on information extraction from tables 3
2.2 Relevant work on text generation 7

II. Corpus Construction for Tables in RCT Articles 14

1. Introduction 14
1.1 Datasets in the open domain 14
1.2 Observation on RCT tables 16
1.3 Proposed annotation tasks 18

2. Methods 18
2.1 Data collection 19
2.2 Information model and annotation guideline development 21
2.3 Annotation 22

3. Results 25
3.1 Information model for RCT tables 25
3.2 Annotation results 29

4. Discussion 32

III. Extracting Structural and Semantic Information from RCT Tables 35

1. Introduction 35

2. Methods 37
2.1Datasets 37
2.2Recognizing table structure 38
2.3Recognizing entities in headers in RCT tables 42
2.4Recognizing values in data cells 44

3. Results 47
3.1Results of recognizing table structure 47
3.2Results of recognizing entities in headers 47
3.3Results of recognizing values in data cells 49

4. Discussion 50
IV. Text Generation From RCT Tables 54
1. Introduction 54
2. Methods 56
2.1 Dataset 56
2.2 Overview 56
2.3 Message 58
2.4 The rule-based NLG system 59
2.5 Hybrid method by integrating deep learning 68
2.6 Experiment and evaluation 71
3. Results 75
3.1 Descriptive statistics of the dataset 75
3.2 Results of content selection 76
3.3 Results of different text generation methods 77
3.4 Results of human evaluation 80
4. Discussion 81
V. Conclusion 86
1. Summary of key findings 86
2. Innovations and contributions 88
2.1 Innovations 88
2.2 Contributions 89
3. Limitations and future work 90
4. Conclusions 91
References 92

List of Tables

Table 1 Element role in the structure model. 26
Table 2 Statistics for descriptive text. 29
Table 3 Statistics for structure in annotated 50 tables. 30
Table 4 Statistics for annotated dataset. 30
Table 5 Statistics for values in annotated 50 tables. 32
Table 6 Rules for parsing structure of RCT table. 41
Table 7 Result for parsing of table structure. 47
Table 8 Overall performance of methods for recognizing concepts from RCT tables. 48
Table 9 F1 scores on each concept of methods for recognizing concepts from RCT tables. 49
Table 10 Result for recognition of values from RCT tables on value level. 50
Table 11 Confusion matrix for recognition of values from RCT tables. 52
Table 12 Confusion matrix for recognition of concepts. 53
Table 13 Features for the ranking-SVM 61
Table 14 Types of specification used in the study. 64
Table 15 Description of different methods evaluated in the study. 72
Table 16 Metrics for measuring generated text from RCT table. 73
Table 17 Descriptive statistics of the gold standard texts and the generated texts.. 75
Table 18 The results of the content selection methods. 76
Table 19 BLEU and ROUGE scores for different text-generation methods. 78
Table 20 Example of the generated texts and gold standard text, and corresponding input data by content selection (ranking-SVM) 79
Table 21 Average scores of human evaluation. 81
Table 22 Results of paired t-test between methods for three scores. 81
Table 23 The result of different approaches to represent table.. 84
Table 24 Intra-class correlation scores for human evaluation. 85

List of Figures

Figure 1Exampes of tables in different domains. 16
Figure 2 Workflow for constructing corpus. 19
Figure 3 Workflow of data collection. 21
Figure 4 Example of a pair of table and descriptive text. 24
Figure 5 structure model for RCT table. 26
Figure 6 Example of structure and semantics for a RCT table. 27
Figure 7 semantic model for RCT table. 28
Figure 8 The example of table that has two columns of row headers. 33
Figure 9 Examples of annotation problems. 34
Figure 10 Styles for hierarchical relation between headers in RCT table. 36
Figure 11 An example of RCT table in format of HTML. 38
Figure 12 Workflow of parsing structure of RCT table. 40
Figure 13 Conception representation in BIO format 42
Figure 14 Architecture of model for recognition of concepts in RCT table. 43
Figure 15 Conversion of input by using of structure information. 43
Figure 16 An example that showed the rule based method for recognition of values in RCT table. 45
Figure 17 The architecture of text generation system for RCT table. 57
Figure 18 Details of message object. 59
Figure 19 document plan for descriptive text 62
Figure 20 The specifications used in the study. 67
Figure 21 Table embeddings for the model. 70
Figure 22 Architecture of the DL model for text generation. 70
Figure 23 Different approach to represent a table as input to the DL model. 83

CHAPTER I

Introduction

With the advances of new information technologies, we have entered the Big Data era, where roughly 2.5 Exabytes (10^{18} bytes) of data are being generated every day[1]. Information science, which is primarily concerned with "the analysis, collection, classification, manipulation, storage, retrieval, movement, dissemination, and protection of information"[2], plays an important role in making big data useful (e.g., for decision making). The well-known Data, Information, Knowledge, Wisdom (DIKW) framework[3] presents a typical workflow from raw data to intelligent behavior. During this process, technologies that can automatically generate human-readable textual description from normalized and analyzed data (also known as data-to-text generation) have received great attention, as natural language is the primary communication channel for human beings. Success stories applying natural language generation (NLG) techniques to produce meaningful textual description of real world events have been reported widely, including news articles, weather forecast reports, sport casting commentaries etc.[4–9] These systems can generate documents from data in seconds, thus disseminating findings learned from massive data in a timely fashion.

Similarly, unprecedented amounts of data have been generated in the biomedical domain, due to high-throughput technologies such as Next Generation Sequencing. The bottleneck for biomedical research has shifted from data generation to data management, interpretation, and

communication[10]. Therefore, multiple large-scale efforts have been launched to address this problem. For example, the NIH Big Data to Knowledge (BD2K) Program[11], launched in 2013, aims to support approaches and tools facilitating large-scale data management and analysis in addition to making biomedical data Findable, Accessible, Interoperable, and Reusable ("FAIR")[12]. In the biomedical domain (perhaps in all scientific disciplines), the most common and accessible venue for scientific communication is publications. The number of biomedical publications indexed in MEDLINE has been growing exponentially each year: 410,197 papers were published in 1990, 531,578 in 2000, and 1,259,513 in 2016[13]. Therefore, it is highly desirable to develop systems to assist in article generation from biomedical data, which will greatly improve the dissemination of scientific findings. However, very few studies have investigated issues of data-to-text generation in the biomedical domain.

Here, we propose to study data-to-text generation methods in the biomedical domain. Our long-term goal is to develop methods and systems that can understand biomedical data, both syntactically and semantically, and automatically generate text descriptions that summarize significant findings of the data in natural language. As an initial step, we propose to investigate methods to transform biomedical tabular data into text description in this dissertation work. Tables contain important information of biomedical studies, and articles often contain textual description of significant findings from tables, which makes a great use case for developing data-to-text generation methods in the biomedical domain. Given the complexity of biomedical research and to make the study more feasible, the biomedical articles were limited to those describing Randomized Clinical Trial (RCT) studies. Our hypothesis is that we can develop various informatics approaches to accurately extract information from tables and automatically

generate text to describe main findings from tabular data for RCT studies with a reasonable performance. To achieve this goal, the following specific aims were proposed:

- Aim 1 -- Conduct a manual analysis of tables and corresponding text in RCT literature and develop annotated datasets.
- Aim 2 -- Develop automated methods to extract and normalize table information (e.g., column/row names and values).
- Aim 3 -- Develop text generation methods to summarize major findings from tables in RCT literature.

2. Background

The proposed study primarily consists of two tasks: 1) extracting information from tables and 2) generating descriptive text based on extracted information, which is a kind of NLG problem. In this section, we review relevant work on table information extraction and text generation.

2.1 Relevant work on information extraction from tables

In scientific publications, a significant amount of information is presented in the form of tables. Tables are often used to describe study related data (e.g., experimental results) in a precise and structured format, which makes it easy for readers to capture the information. Although tables might exist in documents of various formats (PDF, text/ASCII, XML/HTML or image), we will focus on mining tables in XML and HTML corpora in this study. In the following sections, we describe relevant work on table detection in documents, table presentation, and information extraction from detected tables in both the general domain and the biomedical domain.

2.1.1 Detection of tables in documents

The first step in automated table processing is to detect tables in documents, recognize functional areas and understand the structural relationships between data in table cells. Recognition of tables in documents in XML and HTML formats may not be straightforward, since the table tags (<TABLE></TABLE>) in markup languages are used to create both 'genuine' (or 'meaningful') tables or 'non-genuine' (or 'decorative') tables that are simply a multi-column layout of a webpage. Various domain-specific heuristic rules[14–16], decision trees[17] and Support Vector Machines (SVM) with composite kernels[18] have been shown to be successful in tackling the table recognition problem. Previous efforts have also been devoted to recognizing basic functional areas (headers, stubs and data cells) in a meaningful table. Wei et al. achieved 93.5% accuracy in detecting the table header and table lines from 276,880 lines of web content from www.FedStats.gov using Conditional Random Fields (CRF) and features like percentage of white space, header features, and different types of characters[19]. Chavan and Shirgave achieved genuine table detection and header recognition by combining a rule-based filter and a C4.5 decision tree with appearance and consistency features, with an overall F-measure of 95.12 on 2697 genuine and non-genuine tables[20]. In terms of other functional areas, Nagy achieved an accuracy of 98.6% for the identification of stub header cells and the start and end positions of value cells on 20 tables, using a linear Bayesian classifier and features representing both character types and cell location[21]. Wang et al. presented a framework that extended structural processing of a table by using an entity detector after header detection. They achieved 90.7% accuracy in detecting headers for 127 tables and 87.3% accuracy in detecting the entity column of 189 tables[22].

2.1.2 Representation of tables

One fundamental step for semantic table analyses is to find a suitable representation of tables of interest. Because of the diversity of table content and the lack of standards in table data representation, various representations have been developed: Hurst introduced an ontological model of a table which captures graphical, physical, structural, functional and semantic aspects[23], which was later modified and adopted by the table information interpretation tool TARTAR[24]. Wang et al. developed a document hierarchy model to represent table structure[25]. Liu et al. proposed a table metadata representation that includes table structure, and layout as well as document backgrounds[26]. Wu et al. represented a web table with a DOM tree and defined DOM tree similarity for tree clustering and information extraction[27]. However, most of these representations are domain-specific and are tailored to various semantic analyses.

Two previous publications have focused on structural representation of tables. Doush and Pontelli defined an ontology of table using Microsoft Excel spreadsheet components and their relationships[28]. The data model of table contains three layers (article, table and cell) and largely extends the spreadsheet ontology[28]. This ontology includes title, header, row, column and several cell types, which differentiate between header and data cells but lack cell types such as super-row cells and stubs. Milosevic et al. presented a table model for computational processing, which consists of table types and a data model[29]. They defined types of table by their dimensionality and, for multi-dimensional tables, whether it is composed of multiple similar tables or not.

2.1.3 Information extraction from tables in the general domain

The ultimate goal for table mining is to understand the semantic structure of a table. The majority of existing studies on table information extraction focus on relational tables on public websites. One fundamental task in understanding relations between table cells is the extraction of attribute-value pairs: Chen et al. presented a rule-based table mining workflow that involves hypertext processing, table filtering, table recognition, table interpretation, and presentation of results. They achieved the extraction of data-value pairs from tables containing airline tour package information, but did not provide any evaluation results[14]. With the help of external databases, further semantic information can be extracted from relations between table cells: Dalvi et al. presented the WebSets tool to extract entity sets and clustered similar entities based on the hypothesis that entities appearing in one table column likely belong to the same concept. They then mapped each entity cluster to a hyponym using a Hyponym-Concept Dataset built from heuristic rules using several other corpora[30]. Muñoz et al. extracted RDF triples from tables on Wikipedia and derived relational semantics[31] by mapping to the reference knowledge base DBpedia[32]. In contrast, Wang et al. achieved knowledge extraction without a reference knowledge base, by defining similarity scores for DOM tree-represented HTML tables and clustering tables based on their similarity[27].

2.1.4 Information extraction from tables in the biomedical domain

There is relatively limited research that focuses on information from tables in the biomedical domain, most of which are for specific sub-domains of biomedicine. Wong et al. used several machine learning classifiers to extract information related to gene mutation (e.g. gene, exon, mutation, codon, and statistic)[33]. Peng et al. mapped information from tables from papers

about genetic quantitative trait locus (QTL) to a pre-defined dictionary or a deep dependency tree, and extracted information to build a soybean QTL database[34]. Luo et al. proposed a model to represent structure of the tables in biomedical literature, and proposed a concept called Connected Value for recognizing a table in PDF documents[35]. Milosevic et al. presented an ontological table model[29] and developed a method to extract patient number, BMI, weight and patient group name from a set of clinical trial tables, with F-measures 83.3%, 83.7%, 57.75% and 71.32% respectively[36], and he then extended the method to recognize adverse reactions from the tables[37]. Shimanina et al. presented a corpus of 500 tables with semantic annotation of table cells from biomedical research papers about human/mouse cancers[38].

2.2 Relevant work on text generation

The task of NLG is to map information from non-linguistic sources into linguistic form (text written in the form of human language)[39]. It requires solving two problems, what to say and how to say[40]. The first question is about understanding the information in the computerized representation and determining what content should be included in the generated text, which often requires some domain knowledge. The second question deals with how to generate current (e.g., following English grammar) and coherent/logical human language. NLG has been studied on various tasks in the real world. For example, successful stories have been reported on generating summary text for sports[5,6,41] and weather news from weather model predictions[7,8]. Recently, more complicated tasks have been looked into, such as generation of poem[42], biography[43], cooking recipes[44] and product reviews[45]. These studies spent significant efforts on improving generated text by considering specific content, structures, orders, coherence and sentiment.

2.2.1 Rule-based text generation approaches

In NLG, the most straightforward way is to use rule and template based methods. Reiter et al. specified a sub-problem of NLG, concept-to-text, which aims to generate language from knowledge sources like databases, expert systems and other forms of knowledge bases[46]. In order to solve the data-to-text problem, Reiter developed an architecture containing three stages: document planning, microplanning, and surface realization, which has been used in many studies[7,8,47,48]. Document planning chooses the content and structure of a document; microplanning decides the lexical choice for content from document planning; and surface realization converts the abstract representation into text and organizes the structure of the text. Sripada et al. developed the SUMTIME-MOUSAM system using this architecture to produce textual marine weather forecasts from numerical weather prediction models, which has been used by their industrial collaborators[7]. The input data for SUMTIME-MOUSAM was a table including wind direction and speed at time points of every three hours. All data were represented as tuples by rows. In document planning, all time points when the wind direction or speed changed from the previous time point were selected as important data to be presented in the output. Then, in the microplanning stage, some lexical templates were generated to describe each item in data tuples and the changes of wind direction and speed. Finally, realization filled all data into templates to generate phrases and decided the order of the phrases. Although this architecture has been widely used in many NLG tasks, it has some limitations, e.g., it cannot be applied to tasks whose inputs are un-processed raw data. The architecture above for concept-to-text was later extended by Reiter et al. in order to solve the data-to-text problem[49]. Compared with concept-to-text, the input of data-to-text usually is un-processed raw data, which requires signal analysis and data interpretation before the document planning stage[49]. Signal analysis

would recognize patterns in the inputs of numerical data, and output discrete patterns and events. Data interpretation would map summarized patterns and events from signal analysis into messages with knowledge from the related domain. The generated messages in this phrase would be sent to document planning for text generation using the above architecture. The architecture has been used in many tasks including those in the biomedical domain, e.g., generating text from raw data obtained from medical devices[50–54].

2.2.2 Machine learning-based text generation approaches

Although rule based methods have been widely used in various tasks of NLG, they are limited by the pre-defined templates and are not scalable to the diverse patterns in human language, especially for large-scale corpora. Besides, it is also difficult for them to dynamically incorporate domain knowledge for text generation. Therefore, different types of machine learning based methods have been proposed to automatically learn the language patterns for NLG tasks, including generative probabilistic models, and neural network based methods.

Usually, records in the table are treated as tuples, including data type, variable and its value. For example, in the task of weather forecast generation, a record could be presented as a tuple of (wind, direction, east)[9]. Use of generative probabilistic models for text generation included three steps: 1) select a series of records to be included in the generated text; 2) for each record, select a series of variables; and 3). select proper words from task-specific vocabularies to integrate the tuples into text.

Due to its success in the image processing area, neural network based methods have also been applied to NLG[6,42–45,55,56]. Kiddon et al. developed a model to generate cooking recipes given the ingredients in the recipes[44]. Specifically, the task was to generate an ordered text

based on a pre-defined agenda. They used a checklist in the model to record whether an ingredient was used or not and adjusted the probability of its occurrence in the generated text. Lebret et al. developed a method to generate biographic sentences given the information of a person in form of a table [43]. They use the n-grams of words and positions of words in the table as input features for the neural network to train a language model. The language model then predicts the probability of the next word given its context words, based on which a sentence is generated. And the released dataset, WikiBio, has been widely used to develop various NLG methods in many studies [57–60]. Yu et al. applied the generative adversarial network (GAN) to NLG[55]. GAN usually has a generator and discriminator with a competitive mechanism between them: the goal of the generator is to generate text that can't be distinguished from human-written text and the goal of the discriminator is to distinguish the generated text from human-written text accurately. The discriminator and the generator are trained together and leading ultimately, to a generator that can generate natural text. Since GAN was originally used for image generation, this method cannot be applied directly for discrete sequence generation. Therefore, this work used rewards functions to replace the loss function to adapt GAN to NLG. Recently, the Transformer based methods have improved the performance in many NLP tasks[61–63]. Transformer made use of self-attention, which can learn long-range dependencies better and be easily parallelized in computation, compared with recurrent neural network[64]. GPT-2 is a pretrained language model that used the decoder architecture of Transformer and it is trained on WebText, a dataset of 40 GB of text. Experiments show that it achieved the state-of-the-art performance in a zero-shot setting[65].

2.2.3 Text generation from image

Another interesting data-to-text work that has been investigated recently is to automatically generate description text for images. Using machine learning (specifically deep neural networks) and large datasets, a number of studies have focused on the generation of image captions. Vinyals et al. presented the Neural Image Caption (GIC) model where images are first encoded to a 512-dimension vector by a deep convolutional neural network (CNN) and decoded to a sentence by Long-Short Term Memory (LSTM) nets[66]. Xu et al. improved the model by replacing the fully connected layer with a lower convolutional layer in the CNN encoder, and incorporated two attention methods (i.e. the stochastic 'hard' attention and the deterministic 'soft' attention) in the LSTM decoder. This model was evaluated on Flickr8k, Flickr30k and COCO dataset using Bilingual Evaluation Understudy (BLEU) and Metric for Evaluation of Translation with Explicit Ordering (METEOR) metrics, and both attention methods outperformed the GIC model in all three test sets[67]. Similar methods have been applied to images in the biomedical domain. Kisilev et al. used a semi-automatic lesion boundary detection method to extract a set of semantic descriptors (e.g. lesion shape, margin, orientation, etc.) from breast sonography images. Using these semantic descriptors and SVM, they were capable of generating radiological lexicon descriptors that constitute a medical report[68]. Shin et al. achieved automated Chest X-rays annotation by employing the CNN-GRU (Gated recurrent unit) workflow of image label generation[69]. Instead of using image labels to train the CNN, they employed joint image/text vectors in the training that utilized information from clustered context labels with domain knowledge.

2.2.4 Text generation in the biomedical domain

In the biomedical domain, there is limited work on data-to-text generation; but researchers have started looking into this area, including studies on generating patients' medical history from structured representation of an Electronic Patient Record, in the form of a semantic network [54,70], nursing shift summaries[50,52,53] and medical reports of cardiological findings[71]. One core component of NLG in the biomedical domain is how to translate input data to semantic representations that incorporate domain knowledge, which can be used for document planning, microplanning and realization. Scott et al. developed a system that could generate a patients' history from chronicles that are data-encoded views of patient histories[70]. They defined six events and 14 relations between events and constructed semantic graphs made from spines, which were focused and related events. Then, similar events in spines were aggregated and described. Hunter et al. developed a system called BT-Nurse[52,53], which could generate nursing shift summaries from the Badger system that records several channels of continuous physiological data. They created an ontology containing medical entities, events and relations between events. Then, they translated data from Badger into their ontology and detected important events with values in the normal range. Medical knowledge was used to enrich the information recorded in the ontology. Finally, classical data-to-text methods were applied to all the information in the ontology to generate text. Recently, the many studies focused on synthesizing electronic health records[72–74], so that medical data could be shared for scientific research (e.g. developing clinical NLP methods) without violating patients' privacy. Some studies focused on conditioned generation, for example, Liu proposed a model that can generate EHRs conditioned on patients' structured medical records, which may also assist physicians in writing [73].

A task related to data-to-text is text summarization. It summarizes the gist from documents and generates extractive or abstractive summaries[75]. Extractive summaries are created by using original words from inputs and abstractive summaries have new text summarizing the original inputs. Output of text summarization can be textual or graphical. Hirsch et al. developed a summarization system that can summarize and visualize the most frequently documented problems of patients[76].

2.2.5 Evaluation of text generation

How to evaluate text generated from automated systems is also an interesting research question. Automatic metrics (e.g. BLEU[77], METEOR[78], Recall-Oriented Understudy for Gisting Evaluation (ROUGE)[79], etc.) and human ranks are two common ways to evaluate the performance of an NLG system. Novikova et al. did a systematic review on evaluation metrics for NLG[80]. They found up to 60% of NLG research used BLEU as evaluation metrics. They also tested various automatic metrics and human ranks on three NLG systems with two datasets. They found that 1) metric performance is specific to the system and data; and 2) none of the automatic metrics always correlate to the human ranks on all aspects. Reiter also discussed evaluation using controls, to compare generated text with human-written text[81]. In most cases, generated text could not outperform human-written text. Furthermore, some studies evaluate NLG systems in real world applications. The generated texts are used in real-world tasks and their validity is evaluated by representative real-world users. For example, BT-Nurse was deployed and used in the hospital for care planning, and the majority of its text generation was found to be understandable, accurate and helpful by nurses[52].

CHAPTER II

Corpus Construction for Tables in RCT Articles

1. Introduction

Generating text from data has received great attention and data-to-text technologies have been developed to produce meaningful textual description for several real-world use cases. One of the first steps for developing data-to-text systems is to build corpora for this task (e.g., pairs of tabular data and corresponding text), as many approaches are based on recent machine learning methods. However, there are few publicly available gold-standard datasets for data-to-text generation in the biomedical domain, which limits applications of data-to-text in this domain. This chapter describes our effort in building corpora for generating text from tables in RCT articles.

1.1 Datasets in the open domain

In the open domain, several public datasets are made available for developing data-to-text techniques, such as the WEATHERGOV[9], ROBOCUP[82], WIKIBIO[43], E2E[83], boxscore-data[84], etc. These datasets usually include pairs of raw data files and corresponding texts. Most of these data are relatively simple, often provided as attribute-value pairs. For example, Figure 1(a) shows a sample of data from E2E dataset[83], which includes five pairs of attributes and values: *name*, *eatType*, *food*, *priceRange* and *familyFriendly*.

(a) **Flat MR**

name[Loch Fyne],
eatType[restaurant],
food[French],
priceRange[less than £20],
familyFriendly[yes]

(b)

	Higher PEEP group (n=445)	Lower PEEP group (n=449)
Demographic and clinical variables		
Men	259/445 (58%)	255/449 (57%)
Age (years)	65 (54–73)	66 (56–74)
Body-mass index (kg/m^2)	25·5 (4·2)	25·6 (4·4)
Bodyweight (kg)	72·5 (14·3)	72·7 (14·8)
ARISCAT score*	41 (34–43)	41 (34–47)
Intermediate (26–44)	346/442 (78%)	331/447 (74%)
High (>44)	98/442 (22%)	119/447 (27%)
Smoking status		
Never	245/445 (55%)	242/449 (54%)
Former	111/445 (25%)	119/449 (26%)
Current	91/445 (20%)	91/449 (20%)
Alcohol status (past 2 weeks)		
None	301/445 (68%)	307/447 (69%)
0–2 units	130/445 (29%)	125/447 (28%)
>2 units	16/445 (4%)	18/447 (4%)
Preoperative tests		
Haemoglobin (g/L)	119 (26)	119 (26)
Creatinine (μmol/L)	61 (53–76)	61 (53–76)
Urea (mmol/L)	9·3 (5·7–13)	9·6 (5·7–14)
White blood cells (×10^9 cells per L)	7 (5·7–8·6)	7 (5·7–8·7)
Preoperative oxyhaemoglobin saturation (%)‡	97 (96–98)	97 (96–98)
Abnormalities on chest radiography	23/329 (7%)	18/360 (5%)

(c)

Demographics
Age, mean (SD), y
Men, No. (%)
Education, mean (SD), y
Unmarried, No. (%)
Living alone, No. (%)
Rented accommodation, No. (%)
Unemployed, No. (%)
Depression
Beck Depression Inventory score[a]
Mean (SD)
Median (IQR)
DSM-IV diagnosis of major depressive disorder, No. (%)
Previous depression, No. (%)
Cardiac risk factors, No. (%)
Hypertension
Diabetes mellitus
Hypercholesterolemia
Obesity
Current smoker
Previous acute coronary syndrome
Family history of acute coronary syndrome

Figure 1 Examples of tables in different domains. (a) Data table from the E2E dataset (b) Table from the paper PMC6682759. (c) Table from the paper PMC6583706.

1.2 Observation on RCT tables

In this project, our goal is to generate a dataset containing pairs of tables and corresponding description text in RCT articles. Figure 1(b) and (c) show two examples of RCT cohort statistics tables, which typically include demographic and clinical characteristics that are relevant to the study (e.g., specific diseases or drugs). It is relatively easy to identify sentences that describe the corresponding table, so that we can link tables to description text. However, our manual review of RCT tables reveals several challenges of parsing RCT tabular data from biomedical articles.

First, compared with tabular data in open domain, RCT tables are not available as structured attribute-value pairs. We primarily rely on PubMed Central to retrieve tabular data and we found tables in RCT articles in PubMed Central are often in the format of HTML or XML, with very

complex structures, such as nested tables, transposed rows and columns, sub-headers of rows or columns. For example, in Figure 1(c), the "mean (SD)" is located behind the concepts in the cases of "Age" and "Education", yet organized into a row as a sub-header of the concept header in the case of "Back Depression Inventory score". Therefore, we need methods to accurately identify table structures, e.g., row headers, column headers, and data cells.

Secondly, column and row headers of RCT tables often cover much broader types of concepts, e.g., diseases, symptoms, drugs, lab tests, and statistic measures. Unlike many datasets in open domain, which have limited attributes (e.g. the E2E dataset includes five types of fields only, (Figure 1a), column or row headers in RCT tables contain concepts with semantic types due to the diversity of biomedical research. As shown in Figure 1(b) and 1(c), the two example tables include different demographic, clinical, and social behavior concepts. Moreover, as these tables are generated by different researchers, different expressions (surface forms) could be entered to express the same meaning, which is known as a lexicon variation issue.

Furthermore, it is also not straightforward to extract values in data cells in RCT Tables. As shown in Figure 1 (b), data cells refer to cells that contain values for a specific row and a specific column. Although data cells may contain single values, most of time a data cell may contain several values and we have to develop programs to parse individual values from one data cell. For example, the data cell "259/445(58%)" in Figure 1(b) contains three values for 259, 445, 58% respectively. We should accurately parse these values from such complex data cells.

There are existing studies that aim to represent and extract structure information from tables, including those in biomedical literature[29,35]. For example, Milosevic et al. proposed a model to represent tables in scientific literature[29]. However, none of the previous work has

developed information models to represent both structure and semantic information of tables in biomedical literature.

1.3 Proposed annotation tasks

Our goal here is to construct annotated corpora for both the information extraction and the text generation tasks described in the following chapters. According to our observation on RCT tables, we propose the following four anntation tasks:

- Extracting table-text pairs by identifying sentences in the full-text articles that describe the corresponding table
- Annotating table structures such as row headers, column headers, and data cells
- Annotating entities in row/column headers with appropriate semantic types
- Annotating values in data cells

We believe such annotated corpora are critical for the proposed data-to-text study in the biomedical domain. This work will also contribute to biomedical informatics by developing a new information model that syntactically and semantically represents tables in RCT publications, with the potential to generalize to other sub-domains of medicine. Moreover, the annotated corpora will be released in the future, which will allow the community to build on our work to extend this important research topic.

2. Methods

Figure 2 shows an overview of the corpus construction workflow, which includes three steps: (1) collecting RCT papers, corresponding tables and texts from the PMC database; (2) developing an information model to represent structural and semantic information of a RCT table through an

iterative process, and creating an annotation guideline; (3) annotating the dataset following the developed guideline. The details of each step will be described in the following sections.

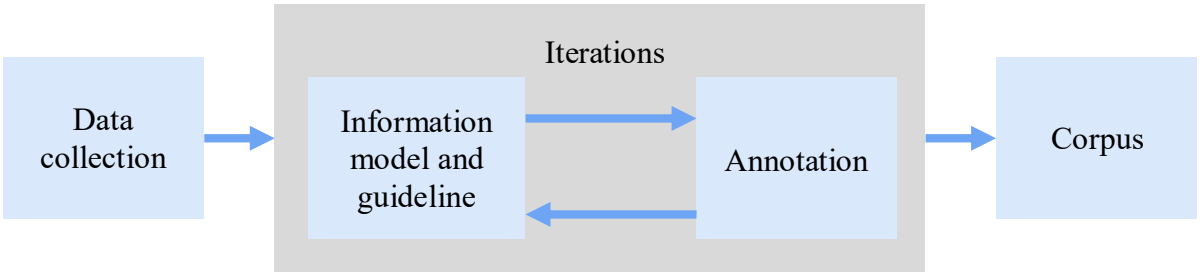


Figure 2 Workflow for constructing corpus.

2.1 Data collection

Figure 3 shows an overview of the data collection process in this study. In order to obtain RCT articles, we queried PubMed using the following criteria: 1) "Publication type" has to be "randomized controlled trial"; 2) limit to four important journals in clinical domain: BMJ, JAMA, Lancet and NEJM, all of which follow the CONSORT guideline [85,86]; 3) limit the publication time from 2011/01 to 2019/01, as CONSORT was released in 2010; and 4) full text articles should be available.

As mentioned in the introduction, unlike tabular data that are used in the data-to-text studies in open domain, tables in RCT literature could be complex, both in structures and semantics. After careful review of different types of tables in RCT articles, we decided to limit our work to baseline tables in RCT papers, which often describe basic demographic and clinical characteristics of different groups in a study. Per recommendation of CONSORT, most of RCT papers included at least one baseline table. Additional inclusion criteria for baseline tables are: a table only has two study arms.

The query to PubMed led to 1,847 papers, and most of them were NEJM (1,032) (Figure 3). Within the 1,847 papers, only 1,048 papers are available on PMC. We randomly selected half of them (518 papers, 1,655 tables) and manually reviewed each article to select baseline tables. Finally, a collection of 279 baseline tables from 277 papers met our inclusion criteria and were used in this study. Most tables are available as XML from PMC Open Access Subset [87]. For papers that don't belong to PMC Open Access Subset, we manually collected the papers and the tables in the format of HTML from the PMC website.

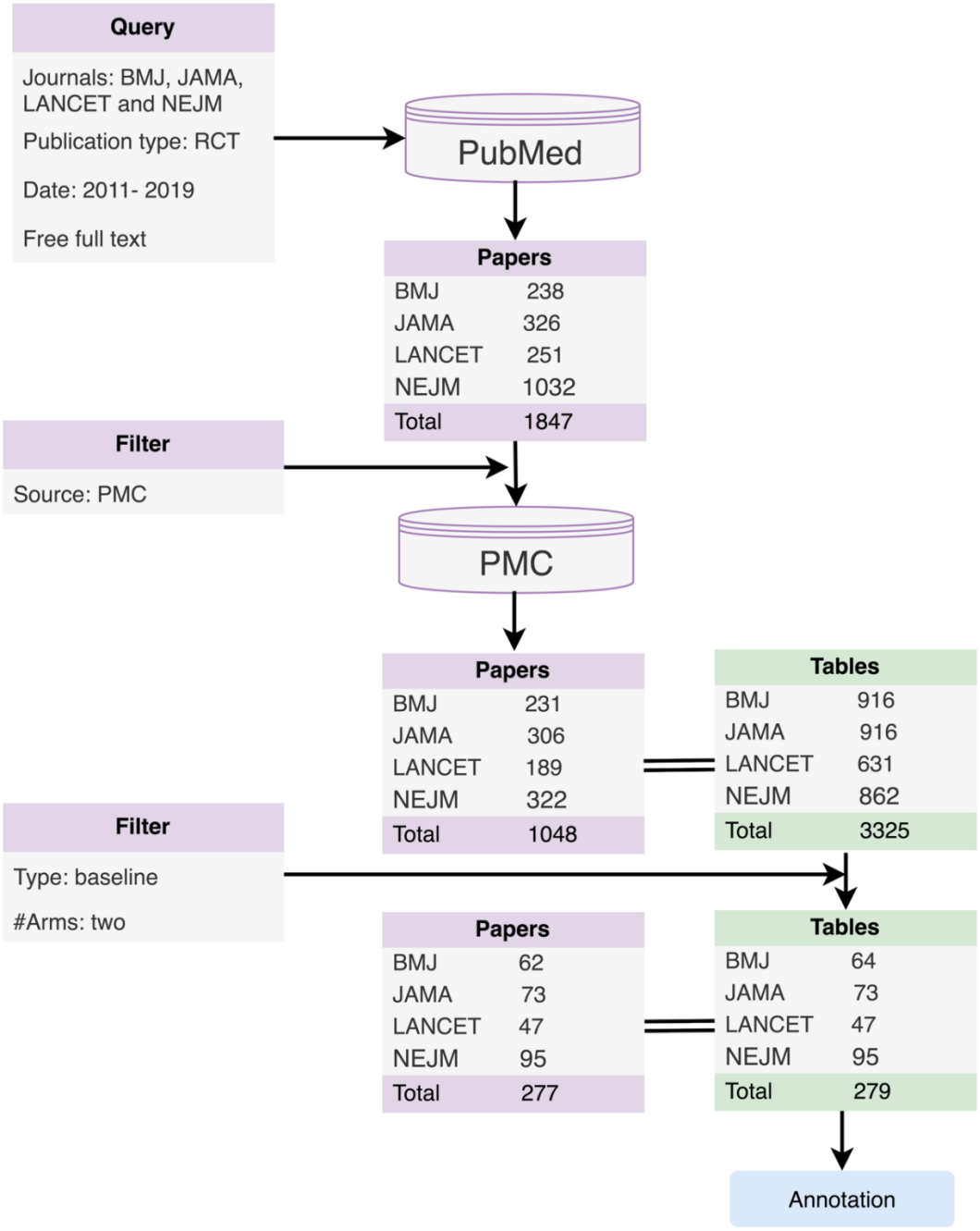


Figure 3 Workflow of data collection.

2.2 Information model and annotation guideline development

Box 1 shows the details of developing the information model for RCT tables. At the beginning we manually analyzed the collected tables to summarize structures of tables, semantic types of concepts in header cells, and values occurring in data cells to propose a raw model. At each

iteration, we first developed a draft guideline based on the information model and conducted a few rounds of trial annotations. Then we analyzed the annotations, discussed discrepancies, and updated the current model and guideline. We repeated these steps and then finalized our table information model and the annotation guideline.

Box 1. Workflow of iterative development of the table model.

1. Propose a raw model M_0 based on manual analysis on a small samples.
2. Iteration:
    2.1 Annotate a small sample with guideline based on M_i.
    2.2 Analyze the annotated samples on:
        - Exceptions that can't be covered by the model
        - Grains of definition of categories in the model
    2.3 Update M_i based on the analysis on 2.2
3. Use M_i as the final model

2.3 Annotation

As mentioned in the introduction, we conducted four annotation tasks: (1) linking pairs of RCT table and descriptive text; (2) structure annotation of tables; (3) semantic annotation of header cells; (4) semantic annotation of data cells.

2.3.1 Linking pairs of table and text

Although it would be great to automatically link tables and description text by recognizing table names in the text, there exist challenges, e.g., one sentence may describe multiple tables and figures. To accurately link tables with texts, we manually reviewed the 279 articles and extracted corresponding descriptive text for each baseline table. As reference text should describe the table

faithfully and accurately without redundant information, we used the following rules to extract descriptive text.

- If a sentence includes text that describe tables and figures other than the target baseline table (e.g. a workflow figure of a clinical study), we remove the text from the sentence. For example, in Figure 4 the texts ("60 patients from 78…", "Patients were …" and "The planned method …") with strikethrough describes Figure 1 and another table (Table 2), so they are removed.
- If a sentence includes a value that is not listed on the table, but it can be inferred by some knowledge (i.e., sum of several values in the table), we keep the sentence. For example, in Figure 4, the percentage number "50%" in green rectangle can be obtained by summing numbers of 23% and 27% under Glasgow Coma Score total, and it's the same for 49%, therefore we keep them in the descriptive text.
- If a sentence includes values that cannot be inferred through information listed in the table (i.e., inference from external knowledge), we remove them from the sentence. For example, in Figure 4, although median ages for two arms are listed in table, the median age for participants overall can't be calculated from them. Therefore, the text in yellow color is removed from the descriptive text.

	Early surgery group (n=305)	Initial conservative treatment group (n=292)
Age (years)		
Median (IQR; range)	65 (55 to 74; 17 to 90)	65 (56 to 74; 23 to 94)
Mean (SD)	63·9 (13·0)	63·9 (13·7)
<60	105 (34%)	106 (36%)
60–69	89 (29%)	70 (24%)
≥70	111 (36%)	116 (40%)
Sex		
Male	174 (57%)	166 (57%)
Female	131 (43%)	126 (43%)
Preintracerebral haemorrhage Rankin*		
0	240 (80%)	236 (81%)
1	41 (14%)	37 (13%)
2	17 (6%)	11 (4%)
3	2 (<1%)	5 (2%)
4	1 (<1%)	2 (<1%)
Preintracerebral haemorrhage mobility*†		
Able to walk 200 m	283 (94%)	275 (95%)
Able to walk indoors	17 (6%)	13 (4%)
Unable to walk	1 (<1%)	2 (<1%)
Glasgow Coma Score, eye		
2	26 (9%)	27 (9%)
3	69 (23%)	65 (22%)
4	210 (69%)	200 (68%)
Glasgow Coma Score, verbal		
1	40 (13%)	44 (15%)
2	36 (12%)	25 (9%)
3	37 (12%)	35 (12%)
4	93 (30%)	96 (33%)
5	99 (32%)	92 (32%)
Glasgow Coma Score, motor		
5	83 (27%)	71 (24%)
6	222 (73%)	221 (76%)
Glasgow Coma Score total		
8	12 (4%)	4 (1%)
9	9 (3%)	15 (5%)
10	23 (8%)	21 (7%)
11	32 (10%)	32 (11%)
12	32 (10%)	34 (12%)
13	46 (15%)	43 (15%)
14	70 (23%)	68 (23%)
15	81 (27%)	75 (26%)

(Continues on next page)

Table 1: Baseline characteristics of patients

Results

~~601 patients from 78 centres in 27 countries were randomly assigned between Jan 11, 2007, and Aug 15, 2012: 307 to early surgery and 294 to initial conservative treatment; recruitment by centre is shown in the appendix p 2. Four patients were excluded because they were recruited by two centres that randomly assigned patients after evacuating the haematoma: a serious protocol violation (figure 1). All other patients were included in the analysis irrespective of the decision of the central CT reading committee about their eligibility (the CT committee's findings will be reported later in a separate paper). This analysis, therefore, includes 305 patients assigned to early surgery and 292 to initial conservative treatment (figure 1).~~ Table 1 shows details of the patients' age, sex, previous medical history, and neuro logical status. The two groups were well matched at baseline. 57% were men and the median age of the patients was 65 years (range 17–94; table 1). ~~Patients were randomly assigned within 48 h of ictus and a quarter (76 [25%] of 305 in the early surgery group and 73 [25%] of 292 in the initial conservative treatment group) were assigned within 12 h (median 21·6 h [IQR 12·0–31·5] and 21·0 [12·0–32·0], respectively).~~ 50% of the patients in the early surgery group and 49% in the initial conservative treatment group had a GCS of 14 or 15 at randomisation (table 1). ~~The planned method of evacuation in 98% of all cases was craniotomy. Table 2 shows the haematoma characteristics reported by site investigators at randomisation. The median volume of the haematoma (with the Broderick method25) was 36 mL (23·0–55·5) and the median depth from the cortex surface was 1 mm (0–2).~~

Figure 4 Example of a pair of table and descriptive text (table and text from PMC3906609).

2.3.2 Annotation of structure, header cell and data cell

We followed the developed annotation guideline to annotate table structures and cells. For table structures and data cells, 50 tables were randomly selected for annotation. For header cells, all 279 tables were reviewed and entities with different semantic types were annotated using the annotation tool in CLAMP [88].

3. Results

3. 1 Information model for RCT tables

The information model is composed of two parts: one to represent structure of RCT tables and the other to represent semantic information.

3.1.1 Information model for structure of RCT tables

Figure 5 shows the model that represents the structure of RCT tables. The model includes five *elements*: *caption*, *row header*, *column header*, *data* and *footer* (Figure 5), and their detailed description is shown in Table 1. In an RCT table, some row/column headers have several sub-headers, and modifiers that are located on the parent headers also apply to their sub-headers. To solve this, we defined a relation type *hasParent*, which represents hierarchical relation between *header*s, *header* and *footer*, or *data* and *header* (Figure 5), where modifiers, statistics and other attributes of parent header can be inherited by its sub-headers. Some elements will have multiple parents. For example, *data* element usually locates on an intersection of a row header and column header, so a data element will have two parents. Figure 6 shows an example of elements (red color) and relations between elements in an RCT table (PMC5533216, Table 1), in which row header “Age at randomization” is parent of both row headers “Median (IQR)” and “Range” .

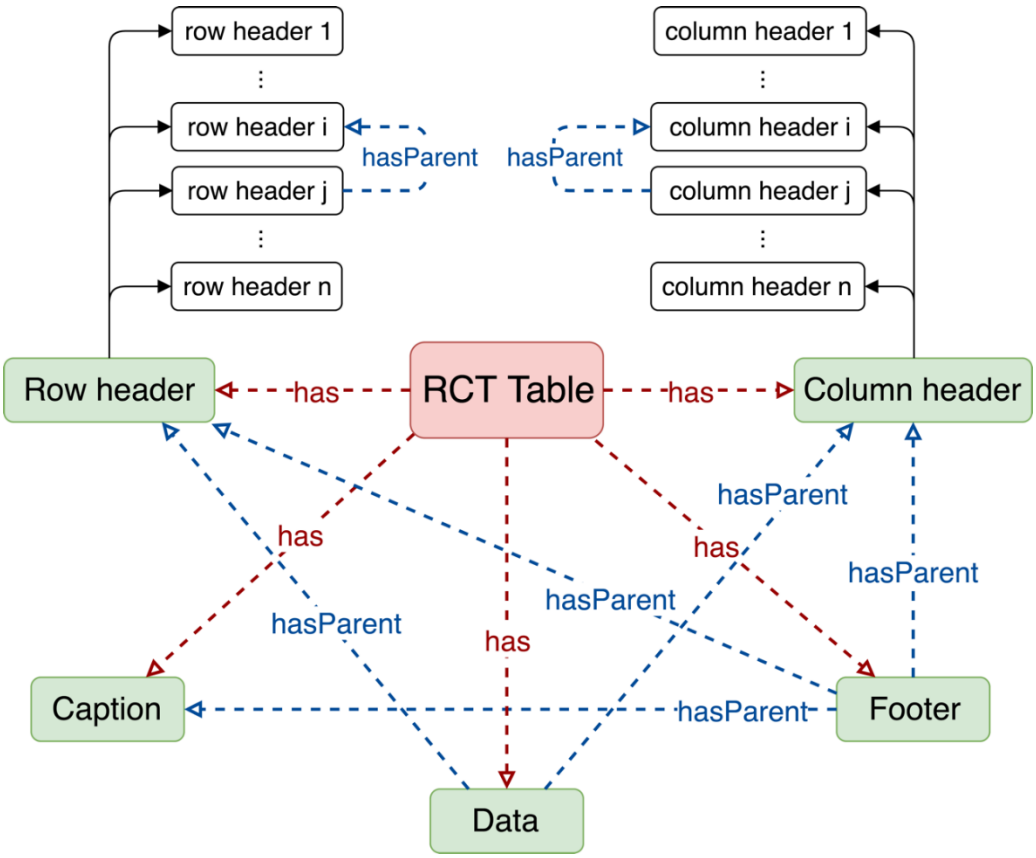


Figure 5 Structural model for RCT table. Solid rectangles represent elements of table, and empty rectangles represent an instance of elements.

Table 1 Element role in the structural model.

Role	**Description**
caption	Caption is at the top of a table, and includes the table id, table title and description of table.
row header	Row header is usually in the first column of each row and includes the attributes of participants in a RCT study and corresponding modifiers and statistical concepts.
column header	Column header is usually in the first rows, and includes arms, statistical analysis for arms.
data	Data is usually an intersection between a row header and a column header, and includes statistical value for some attribute.
footer	Footer is under the table and includes explanation for abbreviations, meaning of values in table or other additional information.

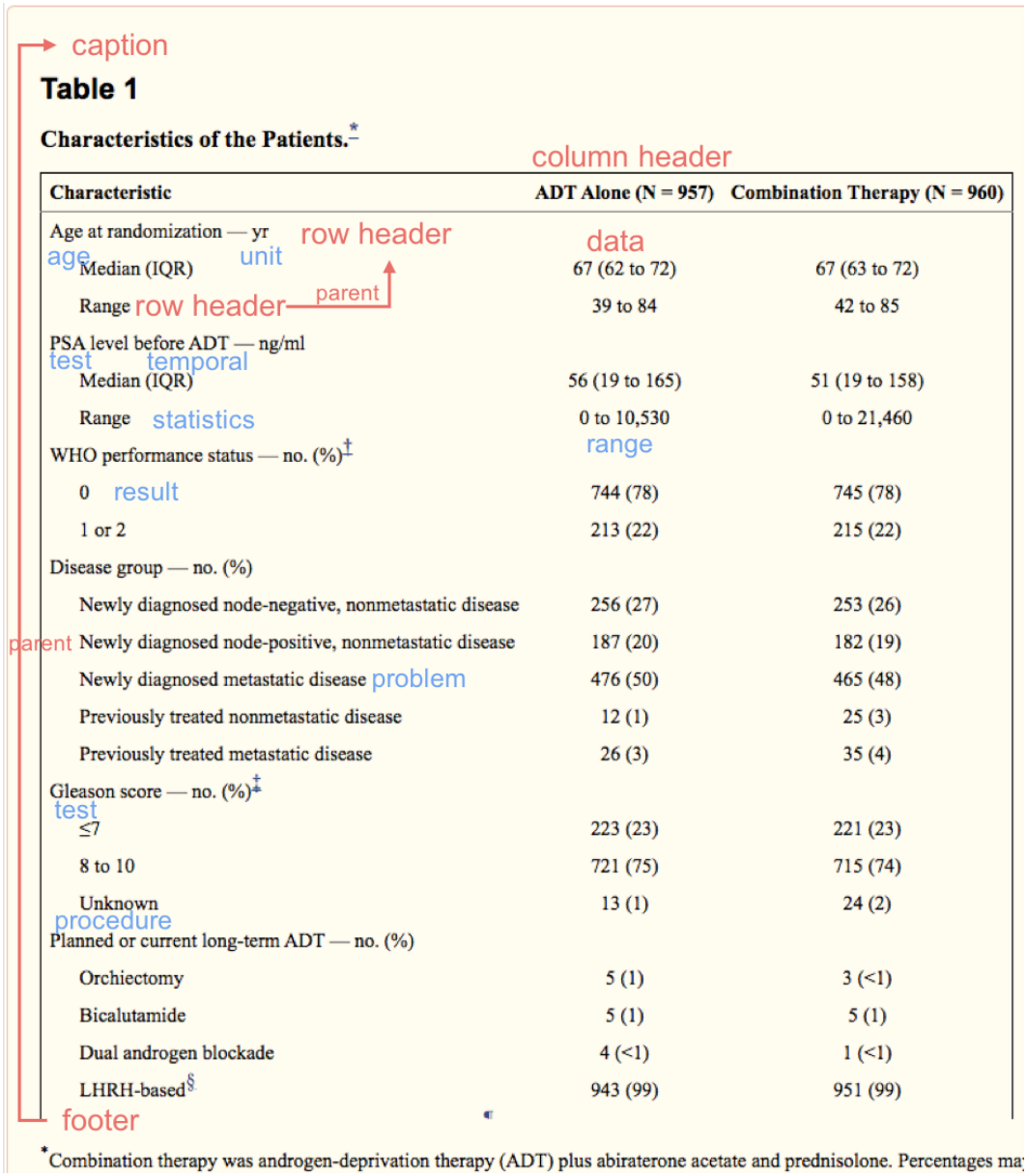


Figure 6 Example of structure and semantics for a RCT table. The table is from the paper PMC5533216.

3.1.2 Information model for semantics in RCT tables

Figure 7 shows the proposed information model for representing semantic information in RCT tables. Table 4 shows the definition for each class. In this model, an *RCT* (pink circle) table can

have multiple *attribute*s (grey circle) and *study arm*s (cyan circle, e.g. “ADT alone” in Figure 6), and *value*s (blue circle) belong to some *study arm* and is possessed by some *attribute*. *Value* can be classified into *single value* and *paired value*. For example, in data “67 (62 to 72) ”, 67 is a single value (median), and (“62” , “72”) is a paired value (IQR). *Attribute* has sub-classes including *demographics* (e.g. “Age” in Figure 6), *clinical characteristics* and *other*, in which *clinical characteristics* includes *drug*, *procedure* (e.g. “Planed or long-term ADT” in Figure 6), *lab test* (e.g. “Gleason score” in Figure 6), *medical problem* (e.g. sub headers under “Disease group” in Figure 6) and *medical event*. Class *other* (purple circle) includes some attributes that can’t be classified into *demographics* and *clinical characteristics*, such as *behavior*, *education*, etc.

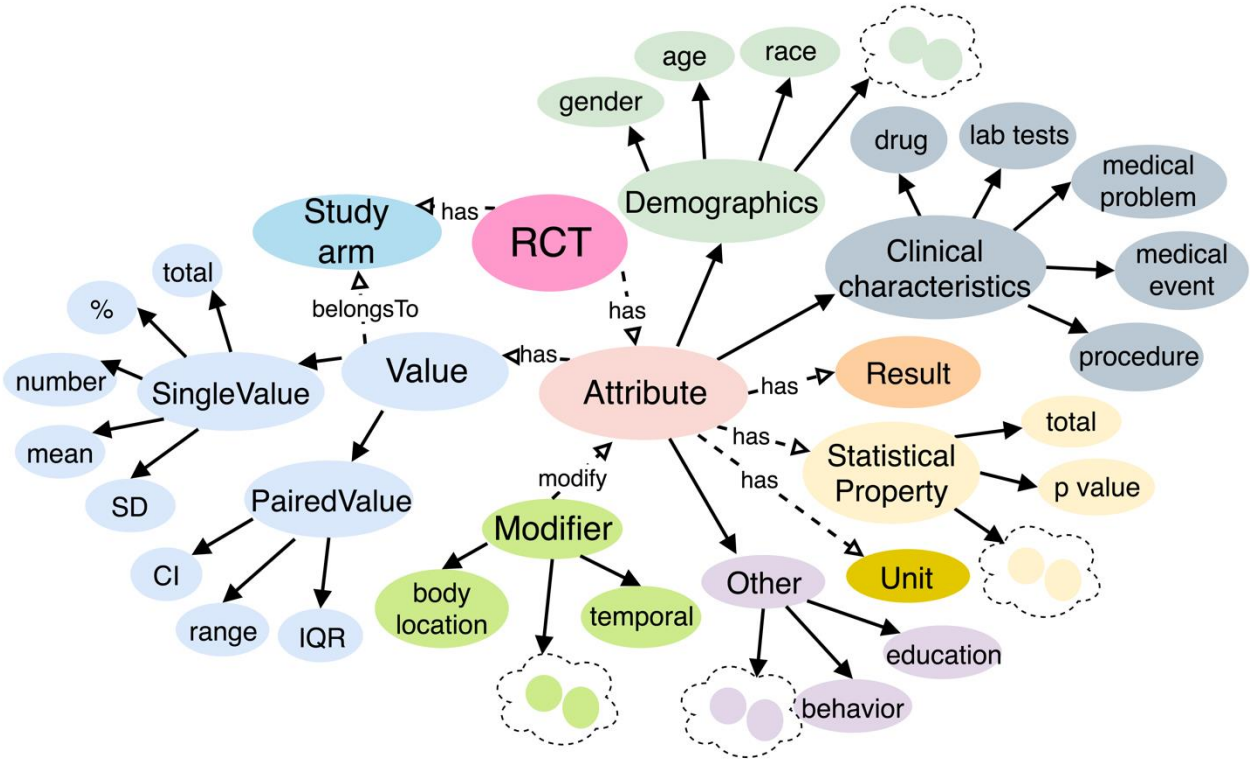


Figure 7 Semantic model for RCT table. Solid circle represents class (concept), solid line represents hierarchical relation between two classes (a class and its sub-class), dashed line represents relation (non-hierarchical) between two classes, and colors are used for distinguishing different classes.

3.2 Annotation results

3.2.1 Pairs of tables and text

Table 2 shows some statistics of the descriptive text in the dataset. There are 450 sentences in the corpus in total, and one table has 1.61 sentences on average. One sentence includes about 21 words on average.

Table 2 Statistics for descriptive text.

Statistics	**Value**
Total number of sentences	450
Average number of sentences per table	1.61
Average length of sentence (words)	21.0

3.2.2 Results of structure annotation

Table 3 shows the results of structure annotation for 50 tables. There are 5,459 elements in total, and most of them are row headers and data cells. There are 35.26 row headers, 2.86 column headers, and 65.96 data cells for each table on average. There are 1,154 relations for row headers (0.65 relations / row header), which means most of row headers are in a hierarchical relation.

Table 3 Statistics for structure in 50 annotated tables.

Element	**Number**	**Average number**
caption	50	1
row header	1763	35.26
column header	143	2.86
data cells	3298	65.96
footer	205	4.1
relations for row header	1154	0.65
relations for column header	10	0.07

3.2.3 Header cell annotation

Table 4 shows the statistics of annotated entities with different semantic types in the dataset. In total, 19 different types of entities were labeled, with 16,700 entities in total. There are more *statistics*, *result*, *problem*, *test* and *unit* entities than other categories.

Table 4 Statistics for annotated dataset.

Category	Sub-category		Description	Count
attribute	*demographics*	*age*	-	543
		gender	-	453
		race	-	687
	clinical characteristics	*drug*	-	551
		medical_event	It refers to some medical events like randomization, admission, etc.	276
		medical problem	-	1,518
		procedure	-	434

		lab test	-	1,374
	other	*education*	-	209
		marital_status	-	168
		behavior	-	115
		other	Attributes that are not clinical characteristics and demographics such as education, behavior, etc.	581
value	*single value*	-	It refers to statistical value that only has one single number like “median”, “mean”, etc.	-
	paired value	-	It refers to statistical value that has a pair of numbers like “range”.	-
study arm	-	-	A group of participants that receive a kind of intervention.	853
modifier	*body location*	-	It refers to some anatomical site.	335
	temporal	-	It refers to a temporal expression that modifies some medical concept above.	131
	measurement	-		180
unit	-	-	Unit of some concept like “mg”, “year”, etc.	1,079
result	-	-	Result of attributes like *label test*, *clinical characteristics* or *other*. It could be range (e.g. “8 to 10” in Figure 6), “Yes”, “No”, “other” and so on.	2,777
statistics	-	-	It refers to statistical concepts.	4,436
Total	-	-	-	16,700

3.2.4 Data cell annotation

Table 5 shows the results of annotation of data cells in 50 tables. There were 6,845 values in total (2.08 values/data cell on average). About 89% of data cells contain single values. Most of the values are *number* (2,259) and *percentage* (2,148), and fewer *pvalue* (35), which indicates that limited studies used a significance test in baseline description. Although *mean* and *sd* are not *paired value*, the numbers of their occurrence are the same, probably because researchers always report them together.

Table 5 Statistics for values in annotated 50 tables.

Item		**Count**	
paired value	iqr_q1	315	760
	iqr_q3	315	
	range_lower	65	
	range_upper	65	
single value	mean	570	6085
	median	331	
	number	2259	
	percentage	2148	
	pvalue	35	
	sd	570	
	total	172	
Total number		6845	
Average number of values per element		2.08	

4. Discussion

In this chapter, we proposed an information model that can represent structural and semantic information of RCT tables and developed an annotation guideline based on the model. We then

built a labeled dataset that can be used for information extraction and text generation, which includes: 1) 279 RCT baseline tables and their corresponding descriptive text; 2) 50 annotated RCT tables with structural information; 3) 279 RCT baseline tables with all row and column headers annotated; and 4) 50 RCT baseline tables with annotated data cells. These annotated datasets will be made publicly available, so that they are valuable not only for this study, but also for other related research projects.

It is worth mentioning that the information model developed for RCT tables could be generalizable to other types of studies in the biomedical domain. As shown in Figure 8, a table for a comparative effectiveness study has two columns of row headers, which was not seen in our dataset. However, it still can be represented by our model, where the first column of row header is the parent of the second of row header, indicating the model's usefulness for other types of studies.

Table 2. Primary Outcome by Treatment Group and Success Rate for the Secondary Comparison of Biliary vs Dual Sphincterotomy for the Pancreatic Sphincter Hypertension Subgroup

	Treatment	No. of Patients	No. (%) [95% CI] of Treatment Success	Risk Difference (95% CI)	
				Adjusted[a]	Unadjusted
Primary outcome	Sham	73	27 (37) [21.6 to 33.6]	−15.7 (−28.0 to −3.3)	−14.3 (−27.3 to −1.2)
	Sphincterotomy (any)	141	32 (23) [15.8 to 29.6]		
Secondary outcome	Pancreatic sphincter hypertension with biliary sphincterotomy	51	10 (20) [8.7 to 30.5]	−10.2 (−27.2 to 6.8)	
	Pancreatic sphincter hypertension with pancreatic and biliary sphincterotomy	47	14 (30) [16.7 to 42.9]		

Figure 8 An example of a table that has two columns of row headers. (Adapted from PMC4428324).

It is challenging to fully represent the semantics of row/column headers. Figure 9 shows some complex examples – instead of being simple or multiple concepts, headers could be complex measures at a given context. We try specifying frequently occurred semantic types and adding them into the model as individual attribute types; meanwhile, we also keep a type of "other" to

refer to all other infrequent types. The decision about the granularity of sematic types in the model is made based on the observation of the data, as well as the specific applications (text generation in this case). Our observation is that description text for baseline tables in RCT articles often mentions statistics about patient demographics and clinical characteristics relevant to the study objectives. Therefore, it is not our intension to fully represent semantic information in header cells. In the future, we plan to leverage other work on this problem, e.g., the Common Data Elements work proposed by biomedical researchers [89].

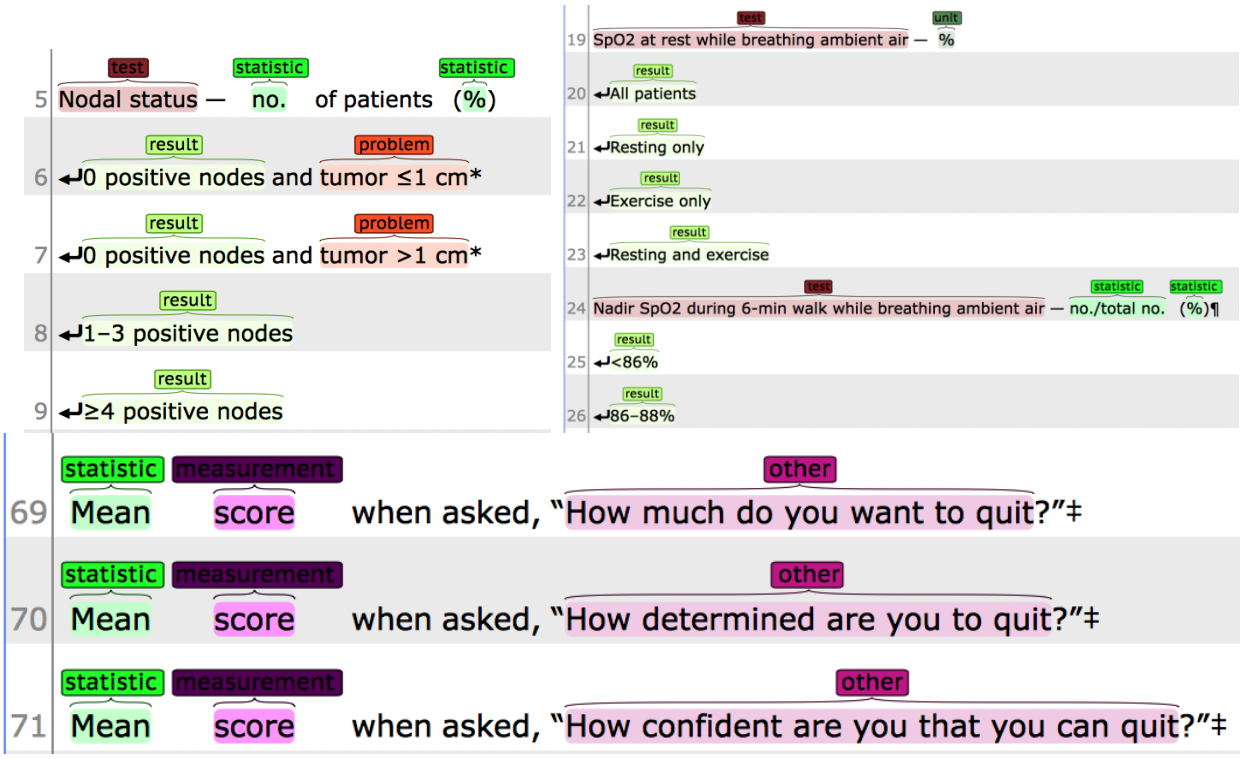


Figure 9 Examples of annotation problems.

Chapter III

Extracting Structural and Semantic Information from RCT Tables

1. Introduction

In this chapter, our goal is to develop methods to extract structural and semantic information from RCT tables, using the annotated datasets built previously. The specific tasks include: 1) recognize structure of tables; 2) recognize entities in header cells; and 3) recognize values in data cells.

As not all the full text articles in PMC are available in the Open Access Subset, we have to develop programs to parse table structures in PMC articles in HTML. Diverse tags and attributes are used in HTML files of RCT articles in PMC, due to different publishers. For example, different patterns of attribute *id* are used to indicate footer element in HTML files such as "<div id="joi180135t1n1" / >" (*id* starting with "joi", from PMC6583083), " " (*id* starting with "tbl", from PMC3898962) and " " (*id* starting with "TFN", from PMC3386296). Moreover, various styles are used to represent hierarchical relations between headers, which brings additional challenges to parse hierarchical relation. Figure 10 a – d shows examples of four styles for hierarchical relations. Some RCT tables use multiple styles to indicate hierarchical relations like (e) in Figure 10, where super row (a super row is a header that takes up a whole row) is used to indicate the hierarchical relation between "Patients' characteristics" and "No (%) of men", and indentions are used to indicate the hierarchical relation between "No (%) by type of stroke" and "Ischaemic". The mixed use of multiple styles

brings difficulty to identify relations between headers, for example, in (d) “Age, years” is at the same level as the super row “Sex”; but in (e) “No (%) by type of stroke” is the sub-header of “Patients’ characteristics”.

(a)

Patient characteristic		
Age (years)	59.8±10.7	60.2±10.3
Male	337 (78%)	332 (76%)
Body mass index	27.8±4.3	27.9±4.5

(b)

Provoked index DVT	142 (56)	161 (62)
Trauma <8 weeks	57 (22)	61 (23)
Surgery (general anaesthesia) <8 weeks	32 (13)	33 (13)
Prolonged immobilisation >6 days	26 (10)	22 (8)

(c)

Demographic	n=662	n=735	n=1397
Mean (SD) age (years); not known	6.3 (0.3); 27	6.3 (0.3); 43	6.3 (0.3); 70
Sex:	n=689	n=778	n=1467

(d)

Sex		
Female	37 (43%)	31 (38%)
Male	50 (57%)	50 (62%)
Age, years	69.4 (13.5)	66.0 (13.2)

(e)

Patients' characteristics			
No (%) of men	82 (65)	80 (65)	0.93
Age (years)	56 (10)	58 (10)	0.32
No (%) by type of stroke:			
Ischaemic	103 (82)	100 (81)	0.82
Haemorrhagic	23 (18)	24 (19)	

Figure 10 Examples of styles for hierarchical relation between headers in RCT table. (a) Uses super row to indicate parent header (PMC6167608). (b) Indentions are used to indicate sub-headers (PMC4886508). (c) Uses bold font to indicate parent headers (PMC3971471). (d) Uses super row and indentions to indicate relation between parent header and sub-headers in one pair of hierarchical relation between for example “Sex” and “Female” (PMC6633921). (e) Multiple styles used to indicate hierarchical relations between headers (PMC3349299).

Entity recognition in table headers is also slightly different from NER in biomedical literature. Table headers are usually short phrases rather than complete sentences in biomedical articles. Therefore, it has limited context around the entities. Although a few previous studies have worked on recognition of concepts from table headers [33,36,37,90] only limited types of concepts such as age, gender, weight, and gene, were extracted in these studies. Moreover, these methods just classify headers to specific semantic types, rather than extracting individual concepts from headers.

Data cells contain values for specific row and column headers. Manuscript authors often use various mathematical symbols and different notations of values, which makes parsing data cells challenging. One particular challenge is having multiple values in one single cell, which is often represented in different formats (for example "6.3(0.3);27" from Figure 10c).

To address these diverse scenarios, we have developed different methods to extract information from RCT tables: rule-based methods are used to recognize table structures and values in data cells and machine learning-based methods are developed to extract biomedical concepts in row/column headers. We believe that our methods developed for recognizing table structures, headers and data cells will be useful not only for text generation from RCT tables in this project, but also for other informatics tasks such as information retrieval from tables.

2. Methods

2.1 Datasets

Here we used the datasets built in the previous chapter, which included 279 pairs of table and text. For structure parsing and value extraction from data cells, we developed rules based on

reviewing tables in a development set and evaluated them on the 50 tables with structure and data cell annotations. For entity recognition in header cells, we used all 279 annotated tables and conducted a 5-fold cross validation.

2.2 Recognizing table structure

Figure 11 An example of an RCT table in HTML format . Green rectangles show some basic patterns for identifying table elements.

Since rule-based methods for recognition of table structure have achieved good performance in previous studies[29,36,90], we took a similar approach to recognize structures of RCT tables in

HTML format (Figure 11) into elements and relations defined in the information model. Figure 12 shows the workflow of the method developed, which includes three steps. In the first step, elements were identified based on some basic patterns (see Table 6 and Figure 12). For example, the text in the HTML tag "<thead>" is identified as *column header*, and the text in the first cell of a row in "<tbody>" is identified as *row header*. Additional rules were applied to fix incorrect recognition (Figure 12). The second step is to recognize indention level of headers, as well as footer text, which can be used for inferring hierarchical relations in the next step. As a table may use different ways to represent indention levels (e.g., super row and indention characters, described in the introduction section, Figure 10 e), some logical indention characters were added before each row headers below super row (highlighted in a red rectangle in Figure 12, and Table 6). Indention characters of each row header are composed of a logical indention character and its original indention character (the short red rectangle in Figure 12). Finally, the indention characters of row headers and footer notion were used to infer its parent.

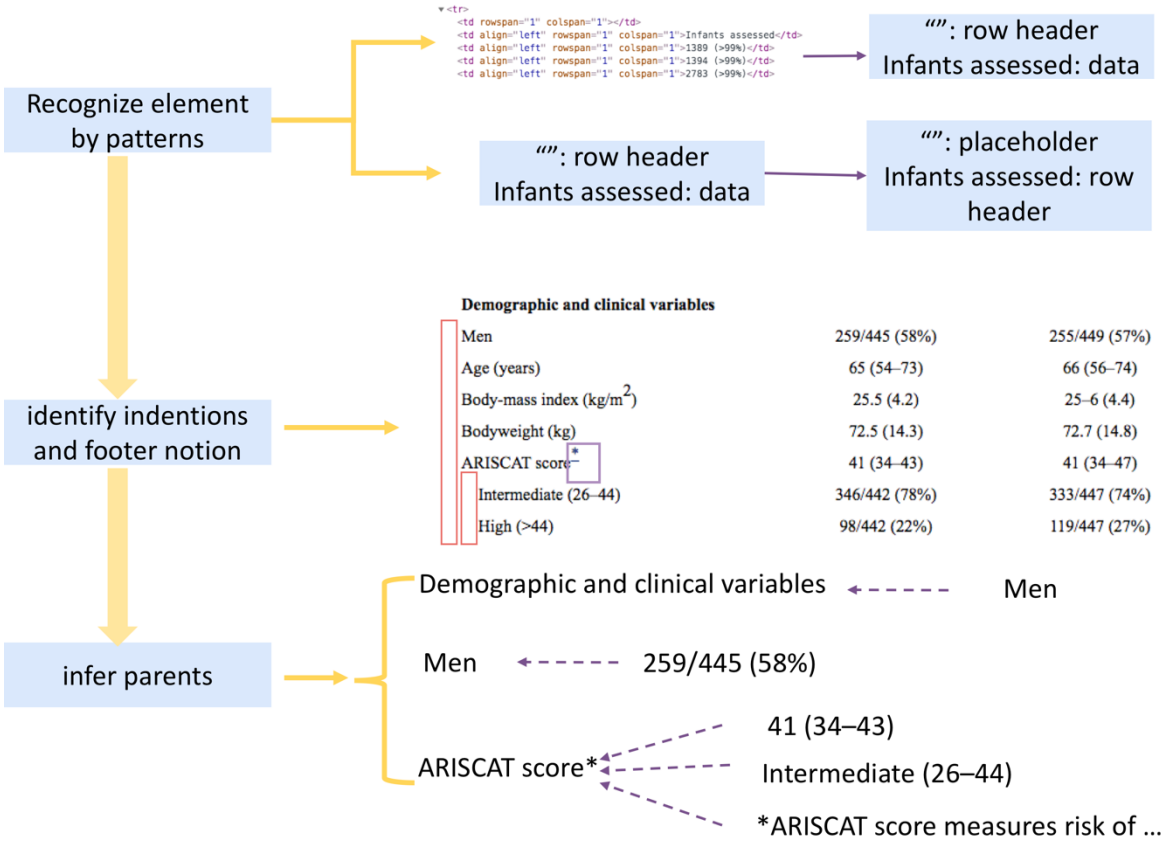


Figure 12 Workflow for parsing structure of an RCT table, which included three steps. Red rectangle represents indention characters, purple rectangle represents footer notion, and purple dot lines represent hasParent relations.

Table 6 Rules for parsing structure of RCT table.

Step	Description of rules
Recognize elements by patterns	Recognize element by basic html patterns: *caption*: … *column header*: <thead>…</thead> *row header*: <tbody><tr>**<td>…</td>** <td/></tr></tbody>, the first td in each row. *data*: <tbody><tr><td>…</td> **<td/> …** </tr></tbody>, tds from the second in each row. *footer*: that is under and has an id starting with "joi", "tbl" and "TFN". Record position (x,y) of column header, row header and data during parsing.
	Fix incorrectly recognized row header: if a row header is an empty <td/>, set the first non-empty element in the row as row header.
Identify indentions and footer	1. Pre-define a set of indention characters (white space characters in Unicode and empty <td/>). 2. If there's a super row in table and there's not any indention character in the row header right below the super row, add a logical indention character to all non-super row headers. 3. Indention characters of a row header equal logical indention and physical indention characters.
	Use the tag <sup> to extract footer notion.
Infer parents	*column header*: given 2 column headers, c_1 (x_1, y_1) and c_2 (x_2, y_2), if $y_1=y_2$ and $y_1+1=y_2$, then c_1 is the parent of c_2. *row header*: Use the indention characters to infer parents. *data*: given a data element, its parents are the closet row and column headers that have the same row and column position respectively. *footer*: given a footer element, if a header has the same footnote as the footer, then it is the parent of the footer element.

Evaluation of parsing of table structure: The rule-based method was applied to 50 annotated tables for evaluation. Accuracy was used for evaluation, where an element is correctly recognized only if both its *element role* and parent relations are correctly recognized.

$$accuracy = \frac{\#correctly\ recognized\ elements}{\#all\ elements}$$

2.3 Recognizing entities in headers in RCT tables

We treat the entity recognition in headers as a sequence labeling task, by converting header text into a sequence using the BIO format (Figure 13), where B- represented beginning of an entity, I- represented inside of an entity, and O represented outside of an entity. The CLAMP system was used for pre-processing steps such as tokenization. Then different machine learning algorithms were implemented and evaluated for this task. To address the issue of lack of context, we also developed a new strategy that integrates information from other cells parsed from table structures.

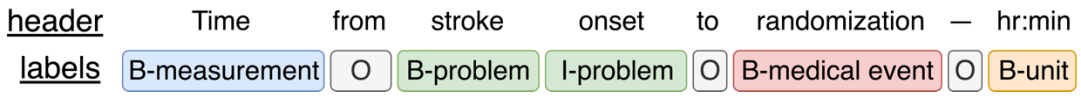


Figure 13 Conception representation in BIO format. The first row was a row header, and the second row was its BIO representation.

Machine learning models: We first included a baseline method for entity recognition using the Conditional Random Field algorithm [91], which has demonstrated good performance prior to deep learning methods [92–95]. Then we also implemented a BERT-based deep learning approach. BERT is a pre-trained context language model that has achieved good performance on many NLP tasks [61,96–98]. BERT is further extended and retrained on biomedical literature (called BioBERT [99]) and we used BioBERT embeddings in this study. Here we adapted BERT to solve the problem of recognition of entities from short table headers. Figure 14 shows the architecture of the model. The input was tokens in headers, which were represented as contextual word embeddings from the BERT model. BIO labels were outputted through a linear and software layer.

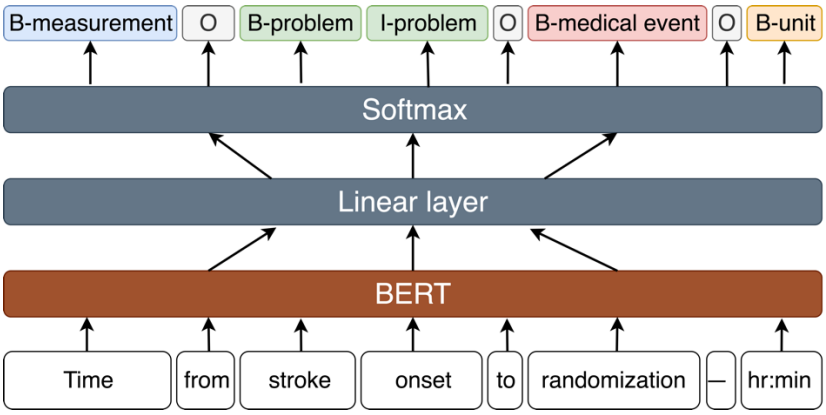


Figure 14 Architecture of model for recognition of concepts in RCT table.

As mentioned in the introduction, headers are usually short text that provide limited context for entity recognition. To address this issue, we developed a new strategy that makes use of other information in a structured table. For example, a row header and its sub-headers could be combined as one long sentence which includes more context about core entities and their related unit, statistics, measurement, modifier and result, thus benefiting the entity recognition model. Figure 15 shows an example of such conversions. We applied BERT to data generated by this strategy and named this approach $\text{BERT}_{\text{strcture.}}$.

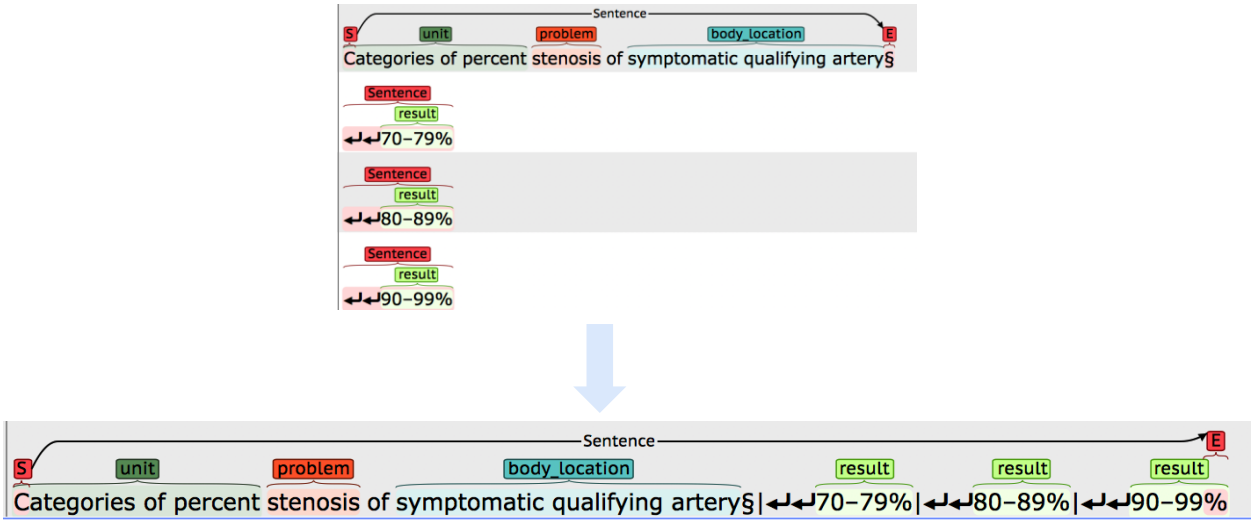


Figure 15 Conversion of input by using of structure information.

Evaluation Three methods: CRF, BERT and $BERT_{structure}$ were developed and evaluated using the 279 annotated RCT tables following a 5-fold cross validation experiment. Standard precision, recall, and F1 score were reported for each method.

2.4 Recognizing values in data cells

A rule-based method was developed to recognize values in data cells of RCT table. Figure 16 shows the workflow of the method developed. Since one data cell may include multiple values, we first split text into values using regular expression. Then certain patterns were identified and rules were developed to determine the types of values, e.g., rule to recognize number type (float or integer), concatenating character for pair value (e.g. ±, /, –, etc.) and other marker strings (e.g. P=, N=, % , etc.). Finally, we used the information in parent headers of data cells to update value types: 1) recognize value types in row header/column header/footer/caption; 2) link values with value types found in headers or footers. Box 2 provides more details of the rule-based method.

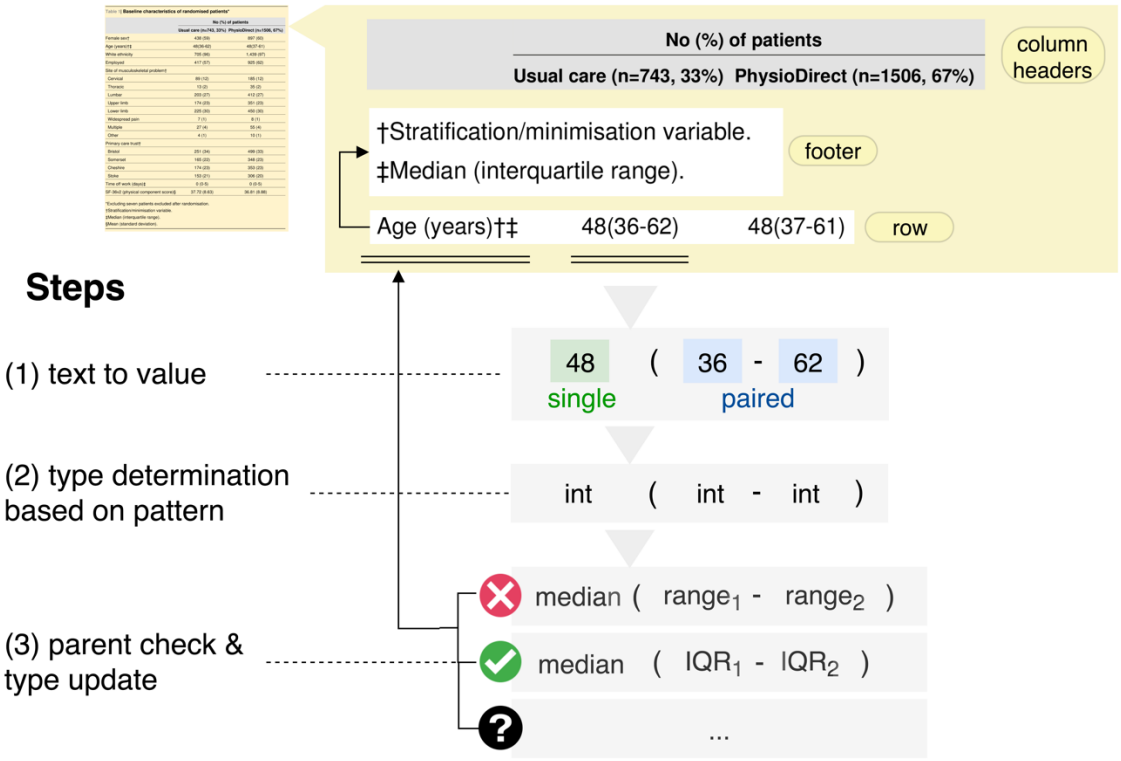


Figure 16 An example that shows the rule based method for recognition of values in an RCT table. It first splits text into values "48" and "36 – 62". Then set a default type median and range for them based on some patterns like integer and "-". At last update their types to median and IQR using information from headers.

Box 2. Rule-based method for recognition of values from data cells.

Step 1: determination of boundary of values from text

1. Define a list of regular expression patterns ***P*** for paired value and single value.
2. Search text using patterns in ***P*** recursively, here is the pseudo-code.

Search (***text***):

for pattern ***p*** in ***P***:

value ← *match*(***text*** , ***p***)

if ***value*** is not None:

break

return ***value*** ∪ *search* (***text*** – ***value***)

Step 2: type determination based on pre-defined patterns

1. Given recognized ***values*** in step 1, define a list of pairs of pattern and corresponding value type $\boldsymbol{P}$= {($\boldsymbol{p_i}$, $\boldsymbol{t_i}$) , …}, and sort them with a predefined priority.
2. for ***value*** in ***values***:
   for pattern ***p*** and type ***t*** in ***P***:
   if ***value*** matches ***p***:
   set $\boldsymbol{value_{type}} \leftarrow \boldsymbol{t}$
   break

Step 3: update value type

1. Use keywords to find value concepts ***VCs*** in *row header*, *column header*, *footer* and *caption*.
2. Link value concepts in *footer* to *row header* and *column header*
3. for ***VCs*** in {*row header*, *column header*, *caption*} in order:
   if ***values*** match ***VCs***:
   update $\boldsymbol{value_{type}} \leftarrow \boldsymbol{VCs}$
   break

Evaluation. 50 annotated tables were used for evaluation, and accuracy, precision, recall, and F1 score were reported. At the element level (i.e., a data cell), an element was scored as correctly recognized only if all values in the element were correctly recognized.

$$accuracy_{element\_level} = \frac{\#correctly\ recognized\ elements}{\#all\ elements}$$

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

3 Results

3.1 Results of recognizing table structure

Table 7 shows the result of parsing RCT table structures. The overall accuracy was 0.9844. Caption and column were correctly parsed with a 100% accuracy and high performance for data cells and row headers was also observed. The performance for recognizing footer was lower (0.8153), probably due to complex tags used for footer in HTML and incorrect recognition of footnote, which led to missing of parent relations.

Table 7 Result for parsing of table structure.

element	Accuracy	Correct	Predict	Gold
data	0.999	3295	3298	3298
row header	0.9773	1723	1763	1763
column header	1	143	143	143
caption	1	50	50	50
footer	0.8153	170	205	212
Overall	0.9844	5381	5466	5466

3.2 Results of recognizing entities in headers

Table 8 shows the precision, recall and F-measure (both exact and inexact matching criteria) for three entity recognition methods: CRF, BERT, and $BERT_{structure.}$ $BERT_{structure}$ achieved the best F1 scores on both exact and inexact match (0.8749 and 0.9260), improving the F1 score around 1.8% for both exact match and inexact match (from 0.8566 to 0.8749 and from 0.9081 to 0.926) compared to BERT, indicating the value of incorporating other structures of from RCT tables.

Table 8 Overall performance of methods for recognizing concepts from RCT tables.

	Exact match			**Inexact match**		
	precision	**recall**	**f1 score**	**precision**	**recall**	**F1 score**
CRF	0.8323	0.8232	0.8277	0.8866	0.8769	0.8817
BERT	0.854	0.8591	0.8566	0.903	0.9133	0.9081
$BERT_{strcture}$	**0.876**	**0.8738**	**0.8749**	**0.9263**	**0.9256**	**0.926**

Table 9 shows the detailed performance of these methods on each type of concept. BERT outperformed others on 4 types of concepts (inexact match), and $BERT_{structure}$ was the best on 13 types of concepts (inexact match). The results for each concept varied widely. Overall ML models performed better on concepts such as “age”, “arm”, “gender”, “race”, “statistics” and “unit”, which usually have less diversity. Model performance on “behavior”, “body_location”, “drug”, “education”, “measurement”, “medical_event”, “other”, “problem”, “procedure” and “test” were lower. For most of them inexact match performance was much higher than exact match performance, indicating that boundary recognition of these entities is challenging.

Table 9 F1 scores for each concept for each method for recognizing concepts from RCT tables.

Concept	Exact match			Inexact match		
	CRF	**BERT**	**BERT$_{structure}$**	**CRF**	**BERT**	**BERT$_{structure}$**
age	0.8585	0.8671	**0.9201**	0.8851	0.9128	0.**9693**
arm	0.905	0.9292	**0.9532**	0.9215	0.9444	**0.9721**
behavior	0.741	**0.7988**	0.7859	0.7672	**0.8498**	0.8446
body_location	0.6337	0.693	**0.7538**	0.7096	0.7887	**0.819**
drug	0.7085	0.7704	**0.7908**	0.8086	0.8759	**0.8873**
education	0.6862	**0.8056**	0.7886	0.8511	**0.9252**	0.9239
gender	0.9604	0.9574	**0.9618**	0.9824	0.9878	**0.99**
marital_status	0.8402	**0.8889**	0.8534	0.9406	0.9485	**0.9698**
measurement	**0.753**	0.7293	0.7252	**0.8133**	0.7721	0.7705
medical_event	**0.7545**	0.7523	0.7518	0.7824	0.7915	**0.792**
other	0.4494	0.4931	**0.5532**	0.5586	0.6257	**0.681**
problem	0.7526	0.8062	**0.8255**	0.8513	0.8928	**0.9108**
procedure	0.652	0.6881	**0.724**	0.7424	0.7867	**0.8234**
race	0.9065	0.9401	**0.9591**	0.9763	0.981	**0.9897**
result	0.7825	0.8229	**0.8821**	0.827	0.8669	**0.9148**
statistic	0.9575	**0.9581**	0.9527	0.9769	0.9754	**0.9776**
temporal	0.7124	**0.7131**	0.6534	0.824	**0.8689**	0.8446
test	0.7605	0.8214	**0.8324**	0.8663	**0.9165**	**0.9272**
unit	0.8757	**0.9198**	0.9165	0.9221	0.9462	0.9393
Overall	0.8277	0.8566	**0.8749**	0.8817	0.9081	**0.926**

3.3 Results of recognizing values in data cells

The overall accuracy for recognizing values in data cells was 0.9044 at the element level (Table 10). The overall F1 score was 0.9098, which was similar to accuracy at element level. The performance for "CI" was 0 because "CI" was not present in the evaluation set of 50 tables; but it

appeared in the prediction of our method. High performance (>0.9) was achieved on most of value types except for “mean” and “range”.

Table 10 Result for recognition of values from RCT tables at the value level.

value type	precision	recall	F1 score
ci_lower	0	0	0
ci_upper	0	0	0
iqr_q1	0.9433	0.8418	0.8896
iqr_q3	0.9326	0.8245	0.8752
mean	0.8296	0.786	0.8072
median	0.9384	0.8278	0.8796
number	0.9283	0.9628	0.9452
percentage	0.9168	0.9646	0.9401
pvalue	1	1	1
range_lower	0.6438	0.7231	0.6812
range_upper	0.6267	0.7231	0.6714
sd	0.9957	0.8175	0.8979
total	1	0.8081	0.8939
Overall	0.9099	0.9098	0.9098

4 Discussion

In this study, we developed multiple methods that can effectively extract information from RCT tables. The rule-based method to parse table structure achieved an accuracy of 0.9844 and the rule-based method to extract values from data cells achieved an accuracy of 0.9044 at the element level and an F1 score of 0.9099 at the value level. Deep learning based methods were developed to recognize entities in headers and the best-performing method achieved an F1 score

of 0.9260. All these results indicate that it is feasible to extract both structural and semantic information from RCT tables with reasonable performance.

There are mainly two types of errors in recognizing table structures. The first one was *missing parents of footer*. Sometimes the footnotes of headers were not surrounded by HTML tag <sup>, which caused failure in recognition. For example, in the footer " *Minimisation variable and predefined subgroup. " (PMC4370502), footnote is not in "<sup>". Another reason for missing parents was that a footer may have multiple parents, which was not considered by our current method. The second type of errors was *wrong recognition of parents of row header*. Some tables used both super row and indentions to indicate relations between headers, which made the proposed method fail to recognize the relations. For example, row header "SF-36 Quality of life" in PMC4447192 does not have a parent, but it wrongly recognized "Demographics" as its parent.

One main error in recognizing data cells is *confusion between paired value* (e.g. *CI* and *IQR*, *range* and *IQR*, and *range* and *CI* in Table 9). For example the value "9-27" (full text: "15 (9–27)", row header: "Marks Asthma Quality of Life Questionnaire (range, 0–80)c,") in PMC5443623, was wrongly recognized as *range*, because it considered "range" as an explanation for type of data. *Confusion also appeared between single value* (e.g. mean vs. number vs. median in Table 11). For example, value "61" (text: "61 (13)") and row header: "Weight (kg)") in PMC3442223 was incorrectly recognized as *number* rather than *mean*, whose parent did not give any explanation for its value type. The explanation for value type was given in footer ("Data are n (%), median (IQR), n/N (%), or mean (SD). ART=antiretroviral therapy."); but it didn't specify which data cells it applied to, respectively, which was impossible for recognizing values. In this example the mean "61" was an integer, which causes that type

number can't be excluded by number type (float or integer), and it requires additional knowledge to know "61 (13)" were *mean* and *SD* rather than *number* and *percentage*.

Table 11 Confusion matrix for recognition of values from RCT tables.

		prediction													
		None	ci_lower	ci_upper	iqr_q1	iqr_q3	mean	median	number	percentage	pvalue	range_lower	range_upper	sd	total
gold	None	0	0	0	0	0	0	0	4	1	0	0	0	0	0
	ci_lower	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ci_upper	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	iqr_q1	1	20	0	266	1	0	0	4	0	0	24	0	0	0
	iqr_q3	5	2	22	0	263	0	0	3	0	0	0	24	0	0
	mean	0	0	0	0	0	448	4	66	52	0	0	0	0	0
	median	0	2	2	0	0	18	274	20	13	0	0	2	0	0
	number	0	5	5	0	0	2	2	2175	66	0	2	2	0	0
	percentage	0	1	1	0	0	38	0	34	2072	0	0	0	2	0
	pvalue	0	0	0	0	0	0	0	0	0	35	0	0	0	0
	range_lower	0	0	0	16	0	0	0	2	0	0	47	0	0	0
	range_upper	0	0	0	0	18	0	0	0	0	0	0	47	0	0
	sd	0	0	0	0	0	34	12	2	56	0	0	0	466	0
	total	0	0	0	0	0	0	0	33	0	0	0	0	0	139

For entity recognition, sometimes drugs were wrongly annotated as procedures (Table 12). Most of the errors had a form of "non-drug + therapy" such as "antithrombotic therapy" in PMC3971471, "previous hydroxyurea therapy" in PMC4358820, "other therapies" in PMC3942158, etc. On the contrary, some drugs in the "drug + therapy" pattern were predicted as procedures (e.g. "warfarin therapy" in PMC3942158).

There was also confusion among concepts of *result*, *medical_event*, *problem* and *test* , as well as between *result* and *other*. Category *other* included all *attribute*s that are not *problem*, *test*, etc.,

and category *result* included results of all *attributes*, so entities in both of them were very diverse, which makes it difficult to differentiate them. From the viewpoint of table structure, *other* usually had some sub-headers that are its result, and as a comparison *result* usually did not have any sub-headers.

Table 12 Confusion matrix for recognition of concepts ($BERT_{structure}$, inexact match).

		gold																			
		age	arm	behavior	body_location	drug	education	gender	marital_status	measurement	medical_event	other	problem	procedure	race	result	statistic	temporal	test	unit	None
prediction	age	519	2	0	0	0	0	0	0	0	1	2	0	0	0	6	0	0	0	0	5
	arm	1	820	1	0	0	0	0	0	0	0	12	1	0	0	1	5	0	0	0	4
	behavior	0	0	144	0	5	0	0	0	0	0	1	8	0	0	8	0	0	2	3	3
	body_location	0	0	0	265	0	0	0	0	0	0	2	15	3	0	35	0	0	5	0	1
	drug	0	6	7	1	502	0	0	0	0	0	6	8	26	0	8	0	0	3	1	11
	education	0	0	0	0	0	190	0	0	0	0	2	0	0	0	13	0	0	0	4	3
	gender	1	0	2	0	0	0	458	0	0	0	1	0	0	0	1	0	0	0	0	3
	marital_status	0	0	0	0	0	0	0	112	0	0	4	0	0	0	1	0	0	0	0	0
	measurement	0	0	0	0	1	0	0	0	136	0	4	3	0	0	8	7	0	3	3	8
	medical_event	0	0	0	0	0	0	0	0	0	216	18	7	5	0	3	0	1	5	0	18
	other	2	8	3	2	0	1	1	1	3	15	395	18	11	1	48	0	0	26	1	42
	problem	0	0	4	10	1	0	0	0	2	9	26	1363	5	0	36	0	0	22	4	18
	procedure	0	1	2	1	23	0	0	0	0	4	13	9	364	0	18	0	1	8	0	6
	race	0	0	0	0	0	0	0	0	0	0	0	0	0	680	1	0	0	0	0	3
	result	9	0	5	24	6	3	0	2	6	0	28	22	9	9	2481	9	5	21	1	19
	statistic	0	12	0	0	0	0	0	0	8	0	0	0	0	0	29	4338	0	5	37	20
	temporal	5	0	1	0	2	0	0	0	0	0	3	0	1	0	1	0	106	0	0	1
	test	0	2	0	1	2	0	0	0	6	4	29	23	10	0	18	2	0	1282	2	13
	unit	0	0	0	1	0	1	0	0	1	0	4	0	0	0	10	22	1	5	991	15
	None	4	3	2	9	15	6	0	0	13	25	42	37	5	1	38	48	13	17	15	0

Chapter IV

Text Generation from RCT Tables

1. Introduction

In this chapter we will develop methods to generate descriptive text from recognized RCT table data, which can be viewed as a data-to-text task in NLG domain. As summarized in related work for the data-to-text task in the first chapter, both classical rule-based methods and more recent deep learning-based methods have been used for the task in many studies. Nevertheless, the specific task here is different from previous open-domain data-to-text tasks in the following aspects.

First of all, RCT tables are more complicated than the datasets used in previous open-domain studies. As discussed in the Introduction section of Chapter 2, RCT tables often have complex structures and cover broad types of biomedical concepts, which makes it is more difficult to summarize/learn patters for both rule-based methods and DL-based methods.

Furthermore, one RCT table usually has tens of concepts and not all of them need to be described in the descriptive text. Therefore, more sophisticated approaches for selecting content to describe are required.

The framework proposed by Reiter et al.[49,100] has been widely used for rule-based NLG systems. However, it requires developing document planning and microplanning components

based on the goal and data of each study. So there are no existing unified approaches and tools to help develop rule-based systems for a new domain such as biomedicine.

Recently DL methods have been used widely for data-to-text tasks and have shown good performance with less engineering effort [43,57–60,83,101]. However, these methods usually require large datasets (i.e., WikiBIO has 728,321 articles). Given that we have only 279 samples in our dataset, it is not clear if DL method alone would work as expected. Lastly, although automatic evaluation metrics such as BLEU and ROUGE have been widely used in NLG tasks [102], they alone are not sufficient to assess the quality of generated text, and human evaluation could be more appropriate if the end goal is use these methods for real world applications [81,103,104].

To address the above challenges, we propose to develop a hybrid approach for RCT table to text generation. We first developed a rule-based system following the classic data-to-text framework proposed by Reiter et al. [100], with customized components built for the biomedical domain. By leveraging some components in the rule-based system (e.g., content selection), we further investigated DL-based approaches for text generation. To reduce the requirement of large data by the DL method, we implemented the following strategies: 1) leverage the content selection component from the rule-based system, and 2) use a pre-trained language model in the DL-based method. In addition to automatic metrics for text generation (e.g., BLEU), we proposed three human-based evaluation metrics. Then a user study was conducted to evaluate the performance of the developed RCT table-to-text systems.

2. Methods

2.1 Dataset

The dataset (279 pairs of table and text) was divided into three sub-sets: training set, development set and test set (3:1:1, 169, 55 and 55 tables for each set, respectively), of which training and development sets were used for training the DL model in the hybrid method (i.e., choosing hyper-parameters of the model), and the test set was used for evaluating both rule-based and DL-based methods.

2.2 Overview

Here we developed two methods to generate descriptive text given recognized concepts and values from RCT tables, a rule-based method and a hybrid method (Figure 17). First, we converted concepts and values extracted from the previous study into *message*s. The *message* is a unit of attributes and values to be described in generated text, which can be easily mapped to linguistic forms later. Usually a message corresponds to a sentence or a phrase [105]. The inputs for the rule-based method are the converted *message*s in a table, and then *message*s were filtered by a content selection component to decide which messages will be kept in the generated description text. In the hybrid approach, the filtered messages are the inputs to the DL-based language generation model. The architectures and components of both systems are described as follows.

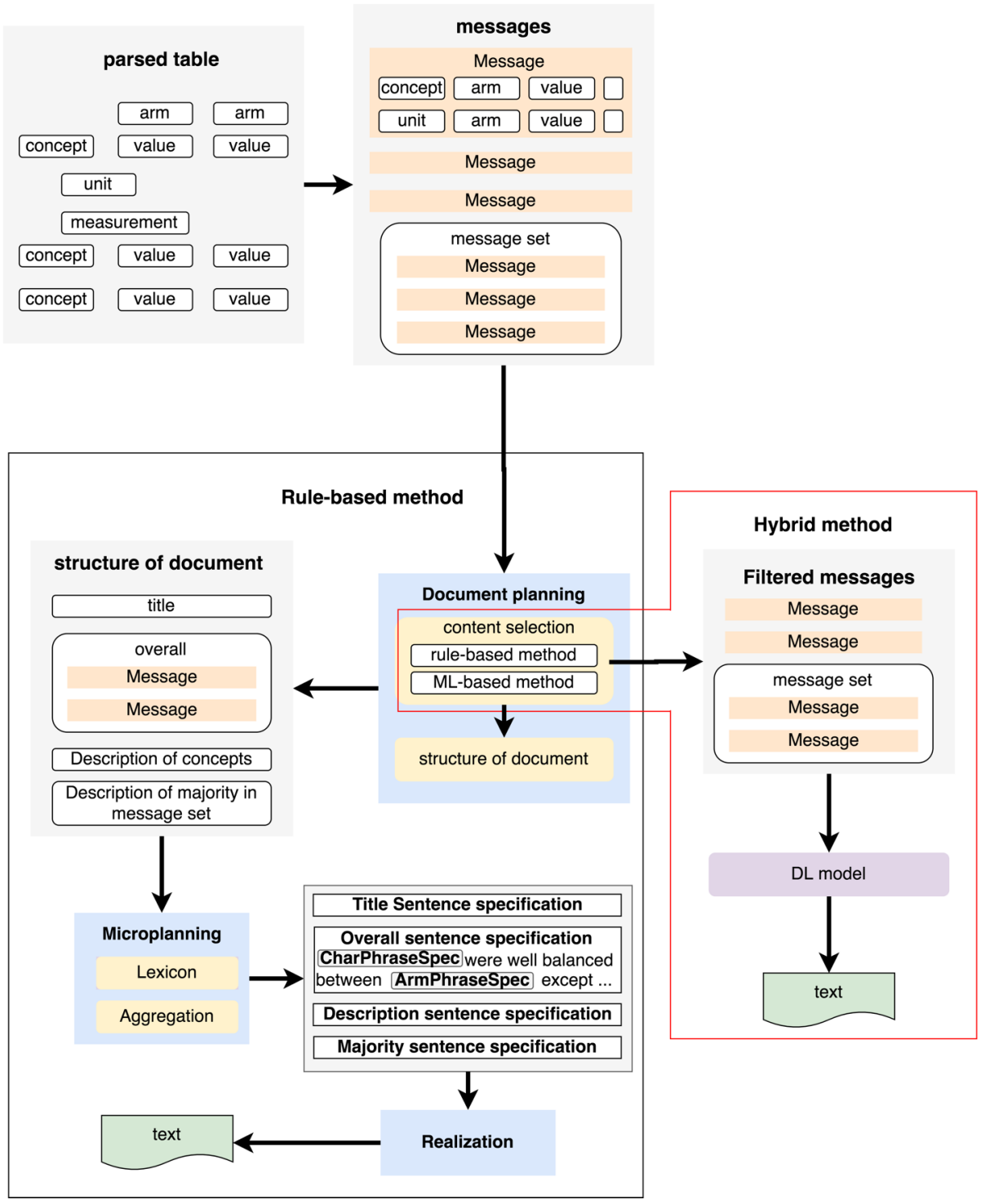


Figure 17 Architecture of the text generation system for RCT table. The recognized tables were converted into messages as input for the rule-based method. The rule-based method included three components (blue solid blocks): document planning component, microplanning component and surface realization component, and the grey blocks represent outputs of the components. In the hybrid method, the filtered messages from the content selection component from the rule-based method were used as input for the DL model.

2.3 Message

Through manually reviewing descriptive text of RCT tables, we found the following frequent patterns of topics: 1) comparing the differences between two arms, e.g., the number of participants for a given condition in each arm; 2) discussing statistical values of some concepts. The message here takes *concept* as the core and also includes all its related information such as values, arms, unit, etc. In this study we defined two types of *messages*: *simple message* and *multiple-result message*.

Simple message is used for representing a concept that does not have extra results (a row header without any sub-headers). Figure 18 shows an example of simple messages. Besides the concept, it also includes other attributes related to the concept such as semantic type of the concept, unit, measurement, arms, and values for arms.

Multiple result message is used for representing concepts (e.g. lab test) that have multiple results. It has the same attributes as *simple message* except that it has an additional attribute *result*, which stands for the results of the concept, and it includes values for arms (Figure 18).

We also named a group of *message*s a *Message Set*, whose concepts are under the same header in a table.

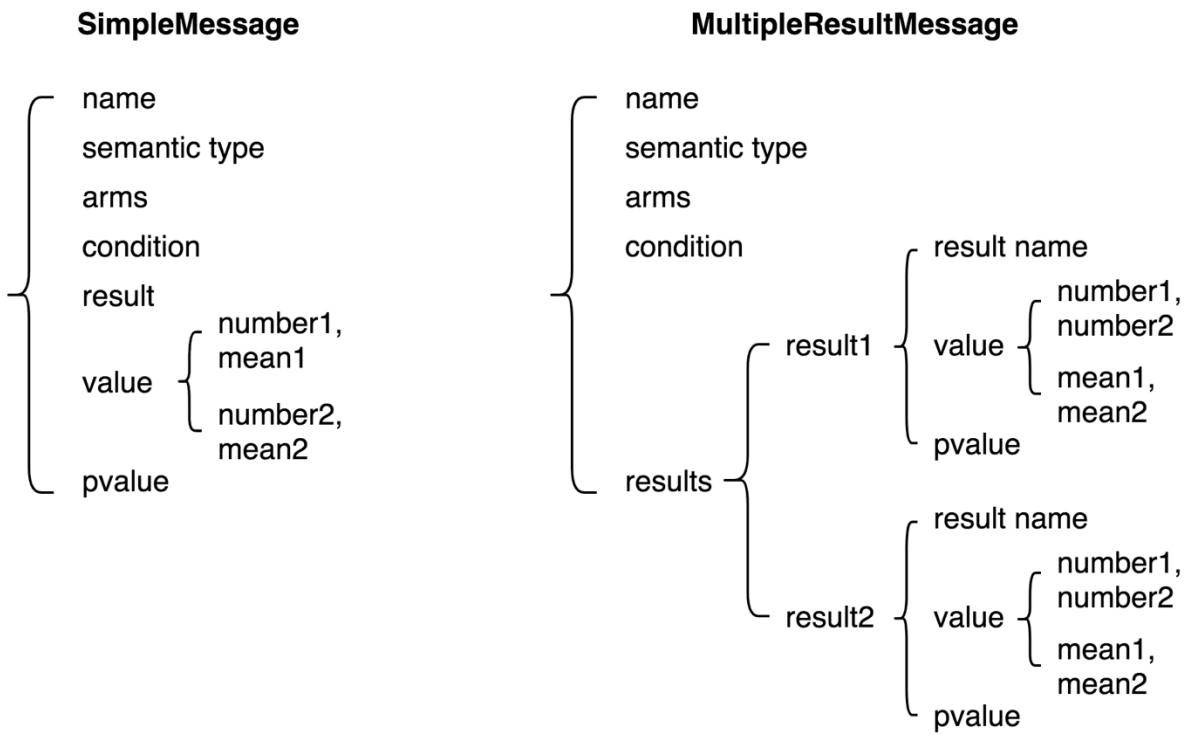


Figure 18 Details of message object. The simple message and multiple result message shares attributes of name, semantic type, arms. The multiple result message can represent multiple results.

2.4 The rule-based NLG system

We followed the data-to-text framework by Reiter et al. to develop the rule-based NLG system. Figure 17 shows the architecture of the system, which includes three components: 1) the *Document Planning* component is to determine which concepts in tables should be described and the structure of generated text; 2) *Microplanning* component is to choose lexicons and syntactic structures for information selected by document planning, and to represent the text as an abstract structure; 3) *Realization* component is to convert the abstract representation into real sentences. The details of each component are described as follows.

2.4.1 Document planning

The inputs to document planning component are *messages*, and its output is a *document plan*. The content selection component is used to determine whether a concept should be described. *Document plan* includes abstract structures of descriptive text

2.4.1.1 Content selection component

A typical RCT table usually includes tens of concepts in headers; but only a few of them are described in text. We developed both a rule-based method and a ML-based method to filter concepts to be described. In order to develop the Ranking-SVM model, we manually annotated which headers in the tables are described.

Rule-based method: Simple heuristic rules are developed, e.g., "if the difference between two arms for a concept is larger than 10%, consider the concept as important".

ML-based method: The concept selection task could also be converted into a ranking problem. The concepts that ranked high will then be chosen for inclusion in the description. In the study, the Ranking Support Vector Machine (Ranking SVM) [106] was used to rank concepts in a table. Ranking SVM is a pair-wise ranking method that ranks candidates for a query by comparing all possible pairs of the candidates. Semantic type of concepts, values of concepts, p value of concepts, position of concepts and a dictionary look-up feature were used as features for the ranking algorithm (Table 13). The categorical feature (semantic type, p value, dictionary look-up) was converted to a continuous value with binary encoding [107]. The categorical variable was first encoded as ordinal and then converted to a binary number, and each digit in the number was viewed as a feature. The package SVM^{rank} [108] was used in the experiment.

Evaluation: A 5-folds cross validation was used to evaluate the performance of the Ranking-SVM. F1-score was reported for different methods. The prediction of the ranking-SVM gave a

rank of all concepts in a table, but it did not decide which concepts should be described. In our experiment, the top *k* concepts in the ranked list were considered as being described (*k* was the hyper-parameter). Additional rules could be added into the ranked list to improve the performance.

Table 13 Features for the ranking-SVM

Feature	Description
Semantic type of concept	semantic type of concept such as age, medical problem, etc.
Values of concept	corresponding values of concept for two arms
Position of concept	position of concept in a table
P value for concept	if p value is not provided, the feature will be set to "UNKNOWN", otherwise the value of the feature will be "TRUE" (p value < 0.05) or "FALSE" (p value >= 0.05).
Dictionary look-up	Because concepts that are related to the study that the table comes from are more likely to be described, the mesh terms and title of the study are viewed as a dictionary. It checks whether a concept is in the dictionary.

2.4.1.2 Structure of the document

Figure 19 shows the document plan for generating descriptive text for RCT tables. The document plan is composed of four parts. The first part is a sentence to describe the caption of an RCT table (e.g. "Table 1 shows the demographic and clinical characteristics of participants."). In the second part, a summary of overall results of comparisons between arms in an RCT table will be described. More specifically, it will state whether the participants between arms are well balanced among the concepts listed in table (e.g., gender, age). It will also state the messages

(concepts), for which participants differed between arms, if applicable. After that, authors usually describe the statistics of a concept (e.g., number of participants or mean of age in an arm). In the last part, we add some description about 'majority' information of concepts. Given a *message set*, the concept that has the greatest number of participants or has a highest mean/median value for some measurements is defined as the majority one.

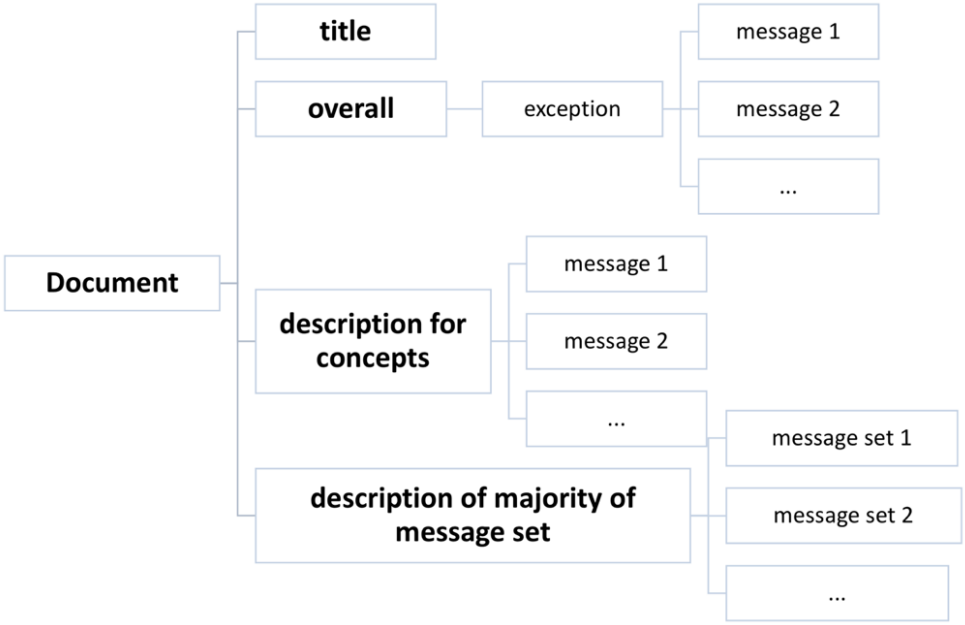


Figure 19 document plan for descriptive text

2.4.2 Microplanning

The goal of the microplanning component is to choose lexicons and produce abstract structures of the text to be generated [105,109]. The output of the microplanning component is *text specification*, which is a data object that clearly specifies the lexicon and structure of the text to be generated. The *specification* could be *canned text*, *abstract syntactic structure* and *lexicalized case frame* [105]. The *canned text* is strings where all lexicons and structures have been already

determined (e.g. the left three examples in Arm Phrase Specification in Figure 20). In *abstract syntactic structure*, syntactic structure of a sentence is specified by a hierarchical structure, whose nodes are syntactic elements (e.g. subject, predicate, object, complement, modifier, etc.) and corresponding lexicons. Additional linguistic features such as tense and voice are also specified here. For example, the second specification in Balance Phrase Specification, is about a sentence "[the characteristics] were well balanced between [study groups].", where the phrases in the brace are variables that can be replaced by other phrases with the same meaning. Because it can use variables and features in specification, *abstract syntactic specification* is more flexible than *canned text*. In *lexicalized case frame*, phrase constitutes are specified by semantic roles rather than syntactic roles, as in *abstract syntactic structure*.

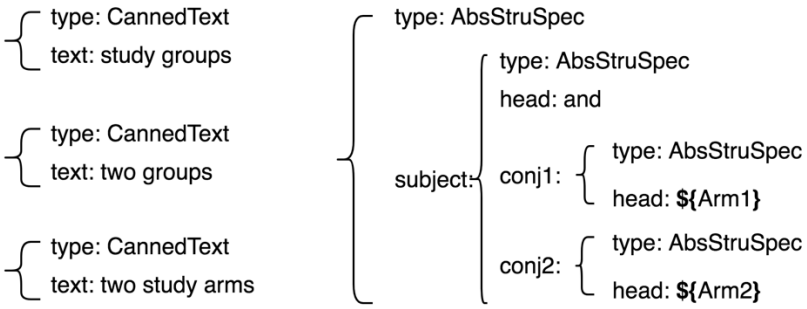
In the study we used both *canned text* and *abstract syntactic structure* to represent the text specification (see details of the specifications in Table 14 and Figure 20). Seven types of specifications were defined in total. In order to make the generated text more diverse, each type of specification included multiple specifications, and Figure 20 shows the details of the specifications in each type. Two of them (*arm phrase specification* and *characteristics specification*) were used for generating phrases needed by the other five types of specifications.

Table 14 Types of specification used in the study. The example shows sentences/phrases that are generated by corresponding type of specification respectively.

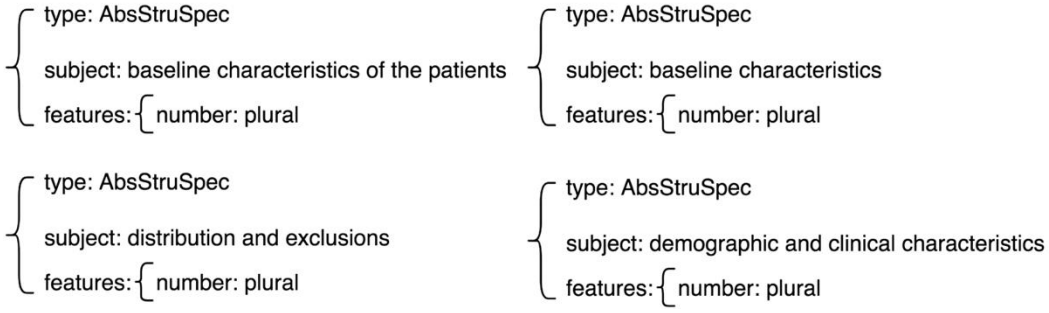
Name of type of specification	Description	Example
Arm phrase	a specification of noun phrase to represent arms/groups	"two groups", "study groups"…
Characteristics phrase	a specification of noun phrase, used as a subject in overall sentence.	"baseline characteristics" …
Title sentence	a specification of description sentence to describe the table.	"table shows the baseline characteristics"
Balance sentence	a specification of sentence to describe that all characteristics are balanced	"no differences were detected between study groups"
Overall sentence	a specification of sentence to describe whether all characteristics are balanced, and if not, which concepts are not balanced.	"the baseline characteristics of the patients were similar in the two groups except the gender"
Description sentence	a specification of sentence to describe value of some concept	"the median age was 64 years"
Majority sentence	a specification of sentence to describe what the majority is in a group of concepts.	"the majority of … was …"

In the microplanning component, messages and message sets in the document plan were converted into corresponding specifications (Figure 17). The second part of the document plan, "overall sentence", included multiple messages that were not balanced. It is longwinded to describe the unbalanced messages separately (e.g. "The age is not balanced. The gender is not balanced. …"). Therefore, the sentences that have the same structure and information (e.g. "The age and the gender are not balanced") need to be aggregated.

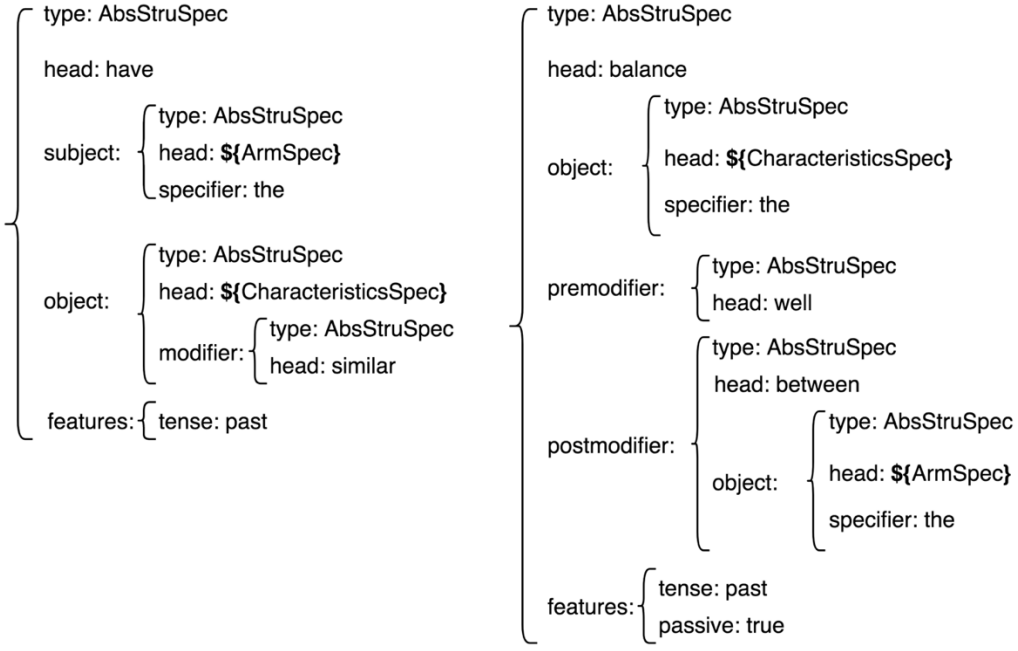
Arm Phrase Specification



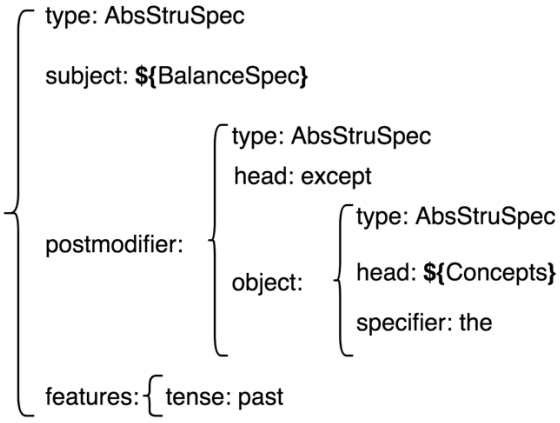
Characteristics Phrase Specification



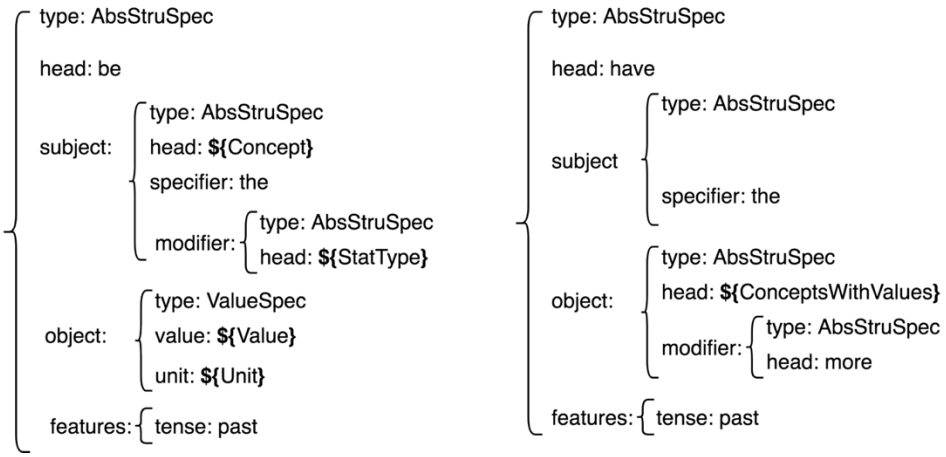
Balance Phrase Specification



Balance With Exception Phrase Specification



Description Phrase Specification



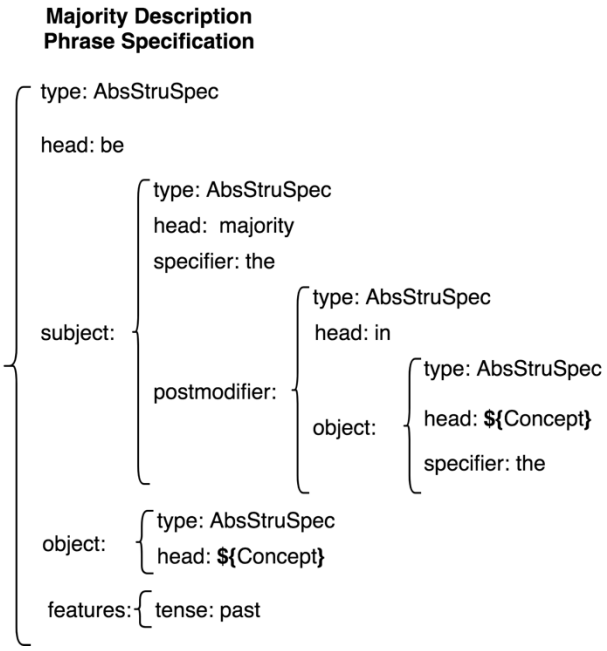


Figure 20 The specifications used in the study. An abstract syntactic specification specified predicate, subject, object, modifiers and features of a sentence. The realizer can choose the correct forms of verbs (e.g. present participle form vs. past particle form) based on the features specified in a specification. Words in a curly brace represent variables that can be replaced by concepts or values in messages.

2.4.3 Surface realization

All abstract representations from the microplanning component will be transformed into sentences according to grammars, called realization. It will first construct a syntactic structure using the abstract representation and then output a sentence based on the syntactic structure. To construct a syntactic structure, the constructor needs to follow English grammar to complete accurate lexicon and structure choices. For example, it should decide the order of multiple adjectives (e.g., “heavy” should be after “large”). Then punctuation and orthography will be further processed to generate a sentence. Some tools have been developed for realization such as KPML [110], OpenCCG [111] and SimpleNLG [112]. SimpleNLG was used in our study, which is a simple and robust realization tool and has been widely used in many NLG tasks. SimpleNLG

allows the use of customized lexicons. In this study, we used the UMLS SPECIALIST lexicon, which is a large biomedical syntactic lexicon and covers general English and biomedical vocabularies from sources such as MEDLINE, medical dictionary, medical books, etc. [113].

2.5 Hybrid method by integrating deep learning

Our hybrid method includes a content selection component from the rule-based system and a DL model for text generation. One advantage of neural network-based text generation methods is that they do not need to explicitly select and organize content and lexicons step by step. In addition, recent advanced deep learning models often can generate more fluent and diverse text sequences.

Essentially, a neural network builds a language model that represents a probability distribution over a sequence of words (denoted as $P(w_1 w_2 \dots w_n)$). Given the words in the 1st to (n-1)th position ($w_1, w_2 \dots w_{n-1}$), the probability of the nth word is denoted as $P(w_n | w_1 w_2 \dots w_{n-1})$. Then the probability over the sequence of words can be calculated by multiplying a series of distribution: $P(w_1 w_2 \dots w_n) = \prod_{i=1}^{n} P(w_i | w_1 \dots w_{i-1}) = \prod_{i=1}^{n} P(w_i | c_i)$, where $c_i = w_1, w_2, \dots w_{i-1}$. In the language model, the probability of the ith word is determined only by its context, c_i. In a data-to-text scenario, the probability of a sequence of words depends on both the context and the input data, and the joint probability of the word sequence could be written as $P(w_1 w_2 \dots w_n | \boldsymbol{C})$, where $\boldsymbol{C}$ represents concepts. Neural network models can learn probability distribution from large training data and generate synthetic texts.

Many studies have developed neural network-based methods for data-to-text tasks [43,57–60,83,101]. Most of these studies train their models from scratch, which often requires a large labeled dataset, such as the E2E dataset [83], WIKIBIO [43], etc. Therefore, it is challenging to develop a good neural network based model to generate text using the small dataset that we have

assembled. Some studies have shown better performance on different NLP tasks by fine-tuning pre-trained language models on an unlabeled corpus of interest [62,63,99], which may provide a potential solution for the data-to-text task in the setting of using smaller datasets. A recent study by Chen et al. followed this idea and proposed the few-shot natural language generation approach, which achieved reasonable performance by employing the pre-trained language model of GPT-2 [114]. GPT-2 [115] is a pretrained language model that used the decoder architecture of Transformer and it is trained on WebText, a dataset of 40 GB of text. Experiments show that it achieved the state-of-the-art performance in a zero-shot setting.

In this study, we adapted Chen's model [114] into the biomedical domain to generate descriptive text from RCT tables. To address the challenges that many concepts are not described in the description text (many negative samples) and the size of our dataset is small, we used the outputs of content selection from our rule-based method as the inputs to the DL model. The *messages* in a table were represented as a group of pairs of attribute and value (Figure 21), and then were converted into embeddings. In Chen et al.'s method, the model used the table embedding method by Lebret et al. [43], which includes embeddings of attributes, words in value, and positions of the words in value (Figure 21). Figure 22 shows the architecture of the model. The outputs of the pre-trained language model was used for calculating attention weights and copy switch p_{copy}, which was used for determining if a word should be sampled from a table or vocabulary. The initial inputs of the pre-trained language model were the encoded table as a context. The encoder encoded the table, whose outputs were used for calculating copy switch and sampling word.

.

arm^a: ultrasound tight control
arm^b: conventional tight control
arm^a^total:118
arm^b^total:112
arm^all^total:230
1^row^concept: women
1^a^value:71
1^b^value:51

Attribute	Word	Position left	position right
arm^a	ultrasound	1	3
arm^a	tight	2	2
arm^a	control	3	1
arm^b	conventional	1	3
arm^b	tight	2	2
arm^b	control	3	1
arm^a^total	118	1	1
arm^b^total	112	1	1
arm^all^total	230	1	1
1^row^concept	women	1	1
1^a^value	71	1	1
1^b^value	51	1	1

Figure 21 Table embeddings for the model. The left is a table represented as a group of pairs of attribute and value, and the right is corresponding embeddings. The arm^a and arm^b represent the names of two arms; the arm^a^total and arm^b^total represent the number of participants in two arms; 1^row^concept is the first concept in the table, and 1^a^value and 1^b^value are the values for two arms. A pair of attribute and value is converted into n embeddings, where n is the number of the words in the value. Each embedding includes embeddings of the attribute, a word in the value and the positions of the word in the value. Position includes left position and right position, and, for example, the word 'ultrasound' is the first word in the value, and the third word in the value from right to left, so the left position and the right position for the word 'ultrasound' are 1 and 3, respectively.

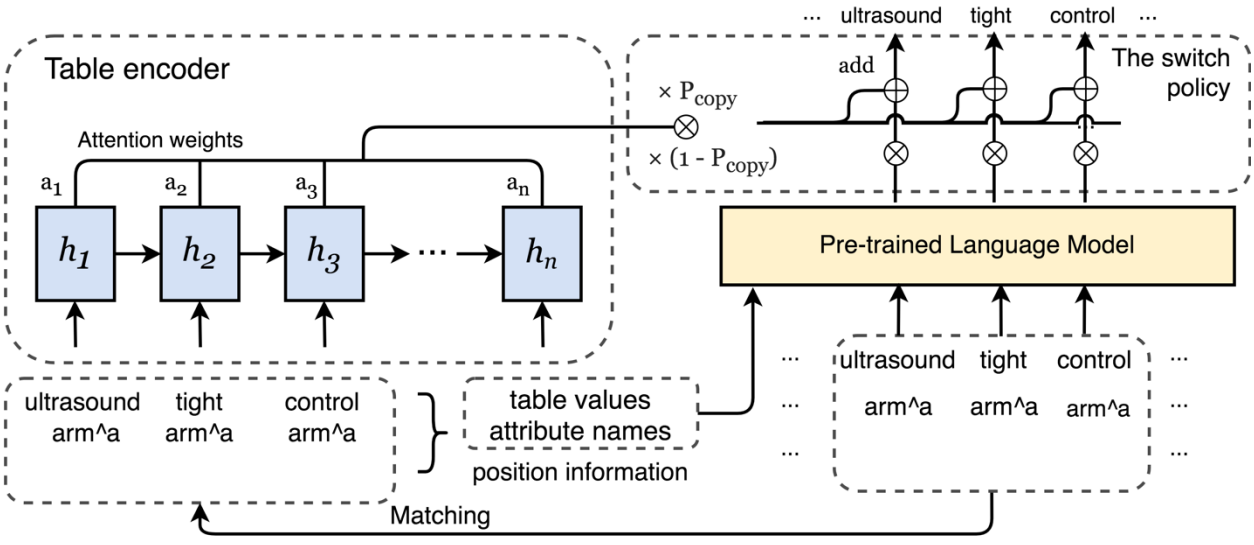


Figure 22 Architecture of the DL model for text generation, adapted from Chen et al. [114]. Pre-trained LM model with biomedical text.

The original pre-trained language model GPT-2 by OpenAI was trained with datasets from open domain. To improve its performance, we fine-tuned the GPT-2 model using texts that describe RCT tables from biomedical literature. We used the following steps to extract texts that describe RCT baseline tables:

- The list of papers with a publication type "randomized controlled trial" from 2011 to 2019 was obtained from PubMed, which resulted in 80,193 papers.
- The full text of these papers were obtained from PMC Open Access Subset, resulting in 33,882 papers.
- For each full text paper, the keywords 'baseline' and 'characteristics' were used to identify baseline table. Then the pattern 'Table ***ID***' (e.g. Table 2, ***ID*** is the ID of the identified table) was used to locate corresponding text in the full text articles. The first sentence that included the Table ID was the start point for text collection. The collected text stopped when a different table or figure ID was mentioned.

The above process led to a set of 7,816 paragraphs describing baseline RCT tables. This corpus was then used for fine-tuning the GPT-2 model (124M).

2.6 Experiment and evaluation

In this study, we primarily investigated rule-based vs. DL-based text generation approaches. Within each approach, we also evaluated different content selection methods: 1) for rule-based text generation, we tested rule-based and RankingSVM-based content selection methods ($Rule_{rule}$ and $Rule_{rank}$), as well as the performance when gold-standard content was used ($Rule_{gold}$); and 2) for DL-based text generation, we tested both gold standard and RankingSVM-based content

selection (DL_{gold} and DL_{rank}). Table 15 provides detailed descriptions of these different experimental settings.

Table 15 Description of different methods evaluated in the study.

Methods	Abbreviation	Description
Rule-based methods	$Rule_{rule}$	The rule-based method was used for content selection, and the rule-based method was used for text generation.
	$Rule_{rank}$	Rank-SVM method was used for content selection, and the rule-based method was used for text generation.
	$Rule_{gold}$	The gold standard data of content selection was used, and the rule-based method was used for text generation.
Deep learning-based methods	DL_{rank}	Rank-SVM method was used for content selection, and the DL-based method was used for text generation.
	DL_{gold}	The gold standard data of content selection was used, and the DL-based method was used for text generation.

Two widely used automatic metrics BLEU [77] and ROUGE [79] were used for evaluation. BLEU and ROUGE measure the precision and F-1 score of n-grams in generated text respectively. The NLTK [116]and the library rouge [117] packages were used to calculate the two metrics respectively. Nevertheless, both BLEU and ROUGE have limitations in evaluating text generation, especially for data-to-text tasks, because there could be many correct ways to describe the data with different lexicons and syntactic structures. Furthermore, it is always important to evaluate generated text for real-world tasks and by actual users. Therefore, we also proposed the following user study to evaluate NLG systems' performance.

Novikova et al. reported a study on human based evaluation for NLG tasks and found that the continuous scale and relative assessments could improve the reliability and consistency of human

ratings [103]. Based on their findings, we defined three metrics: relevance, quality, and matching to measure text generated for RCT tables. The threes metrics are continuous variables with a scale from 0 to 10, and measure whether text describes all relevant information in an RCT table (relevance), the description in text matches the facts in the table (matching), and the grammatical quality of the text (quality). The descriptions, examples and scoring criterion of the metrics are shown in Table 16.

Table 16 Metrics for measuring generated text from RCT table.

Metric	Description and example
Relevance	**Description**: Does the text provide all the useful information from the table? A randomized clinical trial table may have tens of row headers and more than one hundred data cells, and it's unnecessary to describe all the headers and values in the text. Descriptive text only needs to describe cells that contain useful information in the table, such as the demographic or clinical characteristics that are 1) different for the study groups; 2) related to the aims of the RCT study/outcome; 3) other
	Example: *the mean age of the participants was 57 years*. **Explanation**: some characteristics that are more likely to be described, so this characteristic is considered as being useful
	Scoring: For each missed informative point, subtract 0.5 score from total relevance score 10.
Quality	**Description**: The overall quality of the text related to grammatical correctness.
	Example: the two groups were similar with respect to age, sex, race or ethnic group, *baseline number of patients, and baseline number of patients*. **Explanation**: item appears multiple times.

	Scoring: For each gramma error, subtract 0.5 score from total quality score (10).
Matching	**Description**: Is the text accurate and true to the information from the table? 1) Does the text include too much unnecessary information? 2) Is the description not inconsistent with the facts in the table? 3) Does text include numbers or concepts that are not in the table?
	Example: *the control group had more participants* **Explanation:** based on the table the control group has fewer participants.
	Scoring: if the text mentions too much unrelated information, subtract 0-1 score for 1). Each inconsistent point will be subtracted 0.5 score for 2) and 3).

Four users who have medical background were recruited to participate in the human evaluation study. Each user was required to score descriptions from different text generation methods for 20 RCT tables. In total, texts of 40 tables were scored, and texts of each table were scored by two users. For each text, he/she needed to give three scores (relevance, quality, and matching) based on the criterion in the guideline. For each RCT table, we included three texts that were generated by the rule-based method or the DL method, or copied from the original articles (gold standard), respectively, for review. Users did not know the source of the texts. In addition, the display order of the three texts for each table was shuffled, so that the users could not differentiate them by the order. The workflow of the user study is described as follows.

1) There was a half hour training for users to learn the criterion of scoring, and to practice scoring sample texts to be familiar with the criterion.
2) Before scoring a text, users need to read the corresponding table as well as the abstract and title of the paper, together with a summarized list of important characteristics that should be described.

3) Review the three texts, and give scores of relevance, quality and matching for each text.

3. Results

3.1 Descriptive statistics of the dataset

Table 17 shows descriptive statistics of the gold standard dataset and the generated texts from the proposed methods. First, validation and test sets were similar in terms of the number of sentences per text, words per text, and words per sentences on average. Although the number of words per text for generated text by the rule-based method was similar to that in the gold standard set, its sentences were much shorter than that in gold standard set (12.41 vs 21.73 words/sentence in the test set) and each text contained more sentences (2.80 vs 1.71 sentences/text). The text generated by the hybrid method was longer than that generated by the rule-based method.

Table 17 Descriptive statistics of the gold standard texts and the generated texts. The #text, #sent, #word and #word represent number of texts, sentences, words, respectively. The avg_sent_per_text, avg_word_per_text, avg_word_per_sent represent number of sentences per text, words per text and words per sentences on average.

dataset	method	#text	#sent	#word	avg_sent_ per_text	avg_word_ per_text	avg_word _per_sent
training	*Gold*	169	256	5227	1.51	30.93	20.42
valid	*Gold*	55	100	2173	1.82	39.51	21.73
	$Rule_{rank}$	55	146	1896	2.65	34.47	12.99
	DL_{rank}-origin	55	78	1361	1.42	24.75	17.45
test	*Gold*	55	94	2043	1.71	37.15	21.73
	$Rule_{rank}$	55	154	1911	2.8	34.75	12.41
	DL_{rank}-origin	55	91	1563	1.65	28.42	17.18
full	*Gold*	279	450	9443	1.61	33.85	20.98

3.2 Results of content selection

Table 18 shows the results of the content selection methods. It shows that the Ranking-SVM method achieved better performance than the rule-based method. It reached the best performance when semantic types and pvalue were used as features. For the Ranking-SVM method, when the cutoff k was changed to higher numbers (more concepts were labeled as "important"), recall increased greatly; but the precision decreased. The hyper-parameter k=5 was chosen for further text generation because it had a higher recall and F1 score. The post-processing improved the performance from 19.29 to 20.32.

Table 18 Results of the content selection methods. P, R, F means the precision, recall and f1 score. It shows the performance of rank-SVM that used different features.

Method	**Features**		**k=3**			**k=5**			**k=7**		
			P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
RankSVM	**semantic**		14.22	24.19	17.91	12.76	36.18	18.87	10.97	43.29	17.50
	semantic	**pvalue**	15.41	26.22	**19.41**	13.05	36.99	19.29	11.28	44.51	**18.00**
		values	13.26	22.56	16.70	11.04	31.30	16.32	9.58	37.80	15.28
		position	11.83	20.12	14.90	10.11	28.66	14.94	8.96	35.37	14.30
		dictionary	12.90	21.95	16.25	10.82	30.69	16.00	9.53	37.60	15.20
		pvalue + post_processing				15.09	31.10	**20.32**			
Rule			P=6.79, R=41.26, F=11.66								

3.3 Results of different text generation methods

Table 19 shows the results of the rule-based and hybrid text generation methods. All the hybrid methods outperformed the rule-based methods for BLEU-4 and ROUGE-4. For the hybrid methods, DL_{rank} -pmc achieved the best performance (BLEU-4 5.69 and rouge-4 2.44). For the rule-based methods, the performance of $Rule_{rank}$ was higher than $Rule_{rule}$, indicating Ranking-SVM is useful for content selection.

The rule-based method that used gold standard data in content selection ($Rule_{gold}$) did not lead to a significant higher performance than the $Rule_{rank}$ (3.60 vs. 3.07). Similarly, in the hybrid methods the gold standard data did not show a significant improved performance (6.18 for DL_{gold}vs. 4.91 for DL_{rank}), which indicates that the content selection does not affect the scores of BLEU and ROUGE for text generation very much.

The pre-trained language model with PMC text did improve the performance (from 4.91 to 5.69 for DL_{rank}).

Table 19 BLEU and ROUGE scores for different text-generation methods.

	Rule-based methods			**DL-based methods**		
	$Rule_{rule}$	$Rule_{rank}$	$Rule_{gold}$	DL_{rank}	DL_{gold}	DL_{rank}-pmc
BLUE-2	11.38	10.64	13.97	12.04	15.33	13.54
BLUE-3	5.41	5.50	7.33	7.34	9.45	8.50
BLUE-4	2.72	3.07	3.60	4.91	6.18	5.69
ROUGE-1:	23.96	23.09	30.28	26.24	36.06	26.98
ROUGE-2:	5.76	5.74	8.56	9.17	14.05	10.27
ROUGE-3:	1.53	1.83	2.56	3.95	6.81	4.78
ROUGE-4:	0.45	0.69	0.52	2.16	3.49	2.44
ROUGE-l:	24.56	22.94	29.28	26.72	35.6	28.06
ROUGE-w:	11.13	10.24	13.74	13.05	17.9	13.60

As shown in Table 20, the DL methods had the ability to output words from input data including concepts and values (words in red color); but sometimes they generated irrelevant text to the input data, such as “124 patients (47 %)”, in which the numbers didn’t appear in the input data at all. Moreover, because the input data for the DL method did not include semantic types of values, the text generated by the DL method mistook the pre-modifier “mean” as “median” for “age”. On the contrary, the text by the rule-based method correctly chose the pre-modifier “mean” for “age”, because the semantic type was given in its input data. The DL method also mistook the unit for BMI.

Table 20 Example of the generated texts and gold standard text, and corresponding input data by content selection (ranking-SVM). The table used was Table 1 in the paper PMC4267197. The semantic types were only given in the rule-based methods.

Input data by content selection	
arm^a: vitamin d **arm^b**: placebo **arm^a^total**: 161 **arm^b^total:** 161 **arm^all^total:** 322 **1^row^concept:** age **1^a^value**: 47 (mean) **1^b^value**: 48 (mean) **2^row^concept**: bmi **2^a^value**: 27 (mean) **2^b^value**: 28 (mean) **3^row^concept**: women **3^a^value**: 75 (number) **3^b^value**: 75 (number) **4^row^concept**: māori/pacific ethnicity **4^a^value**: 4 (percentage) **4^b^value**: 5 (percentage) **5^row^concept**: baseline serum 25-ohd **5^a^value**: 73 **5^b^value**: 71	
Method	**Example**
Gold	table shows their baseline characteristics.
the Rule-based method	the baseline characteristics were well balanced between two groups vitamin d and placebo except age, woman and māori/pacific ethnicity. the mean age was 47.0 year.

the DL method	the baseline characteristics of the patients were similar in the two groups. the median age was 47 years. 124 patients (47 %) were women. the median bmi level was 27 years.

3.4 Results of human evaluation

Table 21 shows the average scores for human evaluation. Paired t-test was performed to compare these methods on each score (Table 22). The gold standard texts received the best scores by reviewers on two of three measures on average, and overall there is not too much difference between the three groups of texts. The DL_{rank} significantly outperformed $Rule_{rank}$ on scores of grammatical quality. Although both of them used the same content selection component, DL_{rank} had a worse relevance score, probably because the DL model missed some concepts and values in the inputs. Compared with the gold text, $Rule_{rank}$ and DL_{rank} have significant lower scores (Tables 21 and 22). The lower score of $Rule_{rank}$ on match may be because it included more "useless" information in generated text (describing everything from the input data). The lower score of DL_{rank} may be because it included some concepts and values not in the table.

Table 21 Average scores of human evaluation.

Method	Relevance		Quality		Match	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
Gold	9.6	0.7	9.8	1.1	9.8	0.3
DL_{rank}	9.3	0.9	9.9	0.3	9.3	0.9
$Rule_{rank}$	9.5	0.7	9.5	0.6	9.0	0.6

Table 22 Results of paired t-test between methods for three scores. The bold font represents significant p-value (α level is 0.05, with Bonferroni correction).

	Relevance			**Quality**			**Match**		
	$Rule_{rank}$	DL_{rank}	*Gold*	$Rule_{rank}$	DL_{rank}	*Gold*	$Rule_{rank}$	DL_{rank}	*Gold*
$Rule_{rank}$	-	-	-	-	-	-	-	-	-
DL_{rank}	0.0236	-	-	**<.0001**	-	-	0.0352	-	-
Gold	0.3502	0.0279	-	0.0774	0.546	-	**<.0001**	**0.0002**	-

4. Discussion

In this chapter, we proposed two methods to generate descriptive text for RCT tables. The rule-based method achieved a BLEU score of 3.07, and the hybrid method achieved a BLEU score of 5.69. In order to evaluate the validity of the two methods in the real world, a user study was conducted and it showed that two text generation methods could achieve acceptable scores to that of gold standard texts on relevance, grammatical quality and content matching measures.

Content selection remains challenging. Randomness exists even when human (i.e., authors) selects what to describe in the text – different authors may have different choices. For example,

the "Cognitive impairment" in the Table 1 in PMC5710364 is significantly different between two arms, but it is not described in the paper. More in-depth studies are needed to learn how humans interpret tabular data in the biomedical domain.

In the dataset that we developed, each table is associated with only one gold standard descriptive text. In reality, many different ways of describing the tabular data could be valid, as different forms could be used to express the same meaning. However, this makes it difficult to achieve high BLEU or ROUGE scores for the proposed methods. The current rule-based approaches aim to generate diverse text using the modularized specifications. Therefore, it is not its optimization goal to generate "exactly the same text" as the gold standard text.

One problem in the hybrid method that implements the DL model is that they may describe values/concepts that are not in the original table (the hallucination issue) or missed values/concepts that actually appear in the table. For example, in the following text (a generated text), both the concept "women" and the value "65%" were not in the table. One possible reason for the hallucination is that there are only a few values in the gold standard text so that it is difficult for the model to learn.

Example: "… In both groups, approximately 65 % of the patients were women. …"

As shown in the results section, the DL model could not correctly choose pre-modifiers because there is no semantic information for values. Therefore, we examined it and explored different table representations as the inputs of the DL model, including: 1) both semantic types of concepts and values; 2) semantic types of concepts only; 3) semantic types of values only; 4) neither semantic types of concepts or values (Figure 23). Table 22 shows that no obvious improvement was achieved by incorporating the information of semantic types into inputs,

except a slight increase when both semantic types of concepts and values were used. Thus, more investigation is needed into this problem.

- **concept_type-value_type**
  - arm^a:arthroscopic partial meniscectomy
  - arm^b:physical therapy
  - arm^a^total:161
  - arm^b^total:169
  - arm^all^total:330
  - 1^row^age:age
  - 1^a^mean:59
  - 1^b^mean:57

- **concept_type-value_ no_type**
  - arm^a:arthroscopic partial meniscectomy
  - arm^b:physical therapy
  - arm^a^total:161
  - arm^b^total:169
  - arm^all^total:330
  - 1^row^age:age
  - 1^a^value:59
  - 1^b^value:57

- **concept_no_type-value_ no_type**
  - arm^a:arthroscopic partial meniscectomy
  - arm^b:physical therapy
  - arm^a^total:161
  - arm^b^total:169
  - arm^all^total:330
  - 1^row^concept:age
  - 1^a^value:59
  - 1^b^value:57

- **concept_ no_type -value_type**
  - arm^a:arthroscopic partial meniscectomy
  - arm^b:physical therapy
  - arm^a^total:161
  - arm^b^total:169
  - arm^all^total:330
  - 1^row^concept:age
  - 1^a^mean:59
  - 1^b^mean:57

Figure 23 Different approaches to represent a table as input to the DL model. Four approaches are shown. The differences between them are whether to use semantic types of concept and value in attribute of input data (colored in figure).

Table 23 Results of different approaches to represent table. The "concept-value" is used in the study.

	DL methods			
approach name	**Concept_type-Value_type**	**Concept_type-Value**	**Concept - Value_type**	**Concept-Value**
BLUE-2	**16.50**	16.24	15.35	15.33
BLUE-3	**9.72**	9.48	9.02	9.45
BLUE-4	**6.33**	5.79	5.55	6.18
ROUGE-1:	37.09	35.94	**37.72**	36.06
ROUGE-2:	13.79	13.63	**14.2**	14.05
ROUGE-3:	6.55	6.08	5.97	**6.81**
ROUGE-4:	**3.73**	3.15	2.97	3.49
ROUGE-l:	35.72	35.15	**36.21**	35.6
ROUGE-w:	**18.11**	17.76	18.02	17.9

Table 24 shows the intra-class correlation scores[118] (calculated using the package *pingouin*, https://pingouin-stats.org/index.html) for the evaluation. The inter-observer reliability between two users varies a lot on aspects of models, metrics and users. It may be because important characteristics (to be described) defined by user varied a lot, which may cause a lower correlation score especially for the relevance score. Another possible reason is that the users were not well trained. In the future, in order to improve the consistency, the scoring guideline needs to be improved, and workflow optimized to better train users before scoring.

Table 24 Intra-class correlation scores for human evaluation.

	User 1 vs. User 2			User 3 vs. User 4		
	relevance	**quality**	**match**	**relevance**	**quality**	**match**
Gold	0.311	0.117	0.455	0.182	0	-0.603
$\boldsymbol{DL_{rank}}$	0.684	0	0.593	0.611	0.495	0.773
$\boldsymbol{Rule_{rank}}$	0.398	0.107	0.415	0.386	0.301	0.287

Chapter V

Conclusion

1. Summary of key findings

This is an initial study to investigate methods to generate text from scientific tables in the biomedical literature, using RCT tables as a use case. In this study, I proposed an information model to represent both structural and semantic information in RCT tables, built annotated corpora for table structures, semantics, and linked text, and then developed both rule-based and deep learning-based methods for text generation from RCT tables. The key findings for each chapter are summarized as follows.

In chapter 2, I first developed an information model to represent RCT tables. The model consists of semantic classes and their relations to represent both the structure and semantic information in an RCT table. Then we developed a guideline to annotate RCT tables based on the developed information model. A set of 279 RCT tables were collected from the PMC and we annotated the following corpora: 1) linked pairs of 279 tables and corresponding description text in the articles; 2) 50 tables with structural annotation; 3) all 279 tables with annotated entities in header cells (16, 700 labeled entities in total); and 4) 50 tables with annotated values for each data cell. These corpora were later used for developing the methods for text generation.

Chapter 3 describes our approaches to extract structural and semantic information from RCT tables using annotated corpora. Based on our observation, different methods were developed for different types of information: rule-based methods were proposed to parse the structures of RCT

tables, as well as values in data cells; and machine learning and deep learning-based methods were used to recognize entities in table headers. Our evaluation shows good performance for all three tasks, with an accuracy of 0.9844 for structural parsing, an F1 score of 0.9260 for entity recognition in headers, and an accuracy of 0.9098 for value extraction from data cells. To address the issue of limited context of entities in headers, we proposed a new method to integrate other structurally related cells with the target headers and our method improved the entity recognition performance by 1.8% (0.9081 vs. 0.9260).

The methods to generate text from parsed RCT tables were described in Chapter 4. We first developed a rule-based system to summarize the major findings of RCT tables and generated corresponding texts by following the classic framework for data-to-text task [100], which we believe is the first application to the biomedical table-to-text generation. We then investigated the use of deep learning for this task and we realized the limitation of the small dataset that we have. To address this issue, we proposed two strategies: 1) leverage the rule-based content selection component, and 2) implement a new NLG algorithm that is designed for text generation using smaller datasets. Our evaluation using automatic metrics of BLEU and ROUGE shows that rule-based system achieved low performance (BLEU 3.07 and ROUGE 0.69) and DL-based hybrid system achieved improved performance (BLEU 5.69 and ROUGE 2.44), indicating the effectiveness of the DL-based strategies. Furthermore, to address the limitations of the automatic metrics, we further developed human-based evaluation metrics which judge the generated text in terms of its relevance, grammatical quality and matching by human reviewer. A user study by four reviewers was conducted and it showed that the texts generated by the developed methods received lower but close scores to that for the original text in articles, demonstrating the feasibility of our approaches.

2. Innovations and contributions

2.1 Innovations

To the best of our knowledge, this is the first study that attempts to generate descriptive text from scientific tables in biomedical articles (i.e., RCT tables). A number of unique challenges were identified through the study and a series of innovative methods and strategies were developed to address these challenges, including:

- A new information model was developed to represent both structural and semantic information in RCT tables. Although there are a few studies that proposed models to represent the structure of scientific tables in biomedical domain, none of them could represent semantic information of tables. Our proposed model can represent both structural and semantic information of RCT tables, which provides a solid foundation for further text generation.
- An innovative method that can effectively extract biomedical entities from RCT tables was developed. Although numerous studies had been conducted to extract named entities from biomedical text, very few studies worked on extracting entities from biomedical tables, which often lack context information. To address this issue, we proposed a novel strategy to merge relevant table structures (e.g., headers and sub-headers) into one sequence for entity recognition and this shows improved performance. This method could be generalizable to entity recognition from tables in other domains.
- Although a general framework exists for rule-based text generation from concepts, it does not provide specific approaches, tools, or implementations for a specific NLG task. Our rule-based text generation system from RCT tables is one of the first implementations on

generating text for scientific tables in the biomedical domain, which provides guidance for other similar biomedical text generation applications.

- Our hybrid text generation approach that integrates the rule-based content selection and the DL-based few-shot algorithm provides a novel and effective solution for biomedical text generation tasks with limited annotated data, which has not been done in any previous study.
- We also developed new metrics for human-based evaluation for text generation.

2.2 Contributions

This work contributes to the areas of biomedical informatics in the following aspects.

- An information model for RCT tables was developed. It represents both structural and semantic information of scientific tables in biomedical articles and it could be used and extended for representing scientific tables in other biomedical sub-domains (e.g., specific disease areas).
- Annotated tables and linked texts in the biomedical domain were created and will be made available for public use. It serves as a great resource not only for the specific task here, but also for other table-to-text tasks in the biomedical domain, thus accelerating research activities in this area.
- New methods, such as entity recognition for RCT tables and hybrid text generation for smaller datasets, have been developed in this study. They will advance not only table-to-text methods in the biomedical domain, but also similar tasks in other domains.

3. Limitations and future work

As an initial attempt, this study has several limitations. First of all, we limited the scope to baseline tables in RCT studies. Other scientific tables in biomedical literature may have different structures and semantic complexity. Therefore, models developed here may not be directly applied to other scientific tables and additional tuning of the methods is needed when we extend to other types of tables in the biomedical domain. In the future we will expand our studies to other types of scientific tables (e.g. outcome tables of RCT studies), by refining the information model, content selection methods, and text generation methods. Moreover, as discussed previously, the descriptive text in the corpus was collected from original articles, which are subjectively written by individual authors. Text generation models may produce correct description but will not be evaluated favorably using that gold standard data. Therefore, these texts need to be refined by human experts in the future. Furthermore, although the DL model in the hybrid system achieved better performance than the rule-based system, it sometimes generated texts containing concepts and values that are not shown in the source table. This could be a critical issue in cases where accurate information is important. We should develop additional solutions to solve this problem in the future. Lastly, in the study, human based evaluation was conducted, however, the user study was time-consuming and required a large amount of human resources. In the future we will focus on developing automated metrics that can effectively measure the relevance, quality and matching, which can save more resources and be more useful in practice.

4. Conclusion

In this work, I built valuable resources and developed novel methods for information extraction and text generation from RCT tables. The results show that synthetic text generated by our system is comparable to human-written text, indicating the feasibility of this approach. To the best of my knowledge, this is the first study to generate text from scientific tables in the biomedical domain. We believe that the resources (e.g., the information model and annotated corpora) and methods (e.g., the entity recognition from tables and the hybrid text generation approach) developed in this study would be valuable to other biomedical or open domain NLP research and applications.

References

1 IBM. IBM 10 Key Marketing Trends for 2017. 2016. https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WRL12345USEN (accessed 4 Jan 2018).

2 Stock WG, Stock M. *Handbook of information science*. Walter de Gruyter 2013. https://books.google.com/books/about/Handbook_of_Information_Science.html?id=d1PnBQAAQBAJ (accessed 1 Jan 2018).

3 Rowley J. The wisdom hierarchy: representations of the DIKW hierarchy. *J Inf Sci* 2007;**33**:163–80. doi:10.1177/0165551506070706

4 Bastian Haarmann LS. Natural Language News Generation from Big Data. *Int Sch Sci Res Innov* 2015;**9**:1496–502.https://www.waset.org/abstracts/27303 (accessed 1 Jan 2018).

5 Gong J, Ren W, Zhang P. An automatic generation method of sports news based on knowledge rules. In: *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. IEEE 2017. 499–502. doi:10.1109/ICIS.2017.7960043

6 Wiseman S, Shieber SM, Rush AM. Challenges in Data-to-Document Generation. Published Online First: 25 July 2017.http://arxiv.org/abs/1707.08052 (accessed 15 Dec 2017).

7 Sripada SG, Sripada SG, Reiter E, *et al*. SUMTIME-MOUSAM: Configurable Marine Weather Forecast Generator. Published Online First: 2003.http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.8903 (accessed 25 Sep 2017).

8 Reiter E, Sripada S, Hunter J, *et al*. Choosing words in computer-generated weather forecasts. *Artif Intell* 2005;**167**:137–69. doi:10.1016/J.ARTINT.2005.06.006

9 Liang P, Jordan MI, Klein D. Learning semantic correspondences with less supervision. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*. Suntec, Singapore: : Association for Computational Linguistics 2009. 91–9.https://dl.acm.org/citation.cfm?id=1687893 (accessed 19 Dec 2017).

10 Demets D, Tabak L, Altman R, *et al*. Data and Informatics Working Group Report. 2012. https://acd.od.nih.gov/documents/reports/DataandInformaticsWorkingGroupReport.pdf (accessed 1 Jan 2018).

11 Margolis R, Derr L, Dunn M, *et al*. The National Institutes of Health's Big Data to Knowledge (BD2K) initiative: capitalizing on biomedical big data. *J Am Med Informatics Assoc* 2014;**21**:957–8. doi:10.1136/amiajnl-2014-002974

12 Wilkinson MD, Dumontier M, Aalbersberg IjJ, *et al*. The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data* 2016;**3**:160018. doi:10.1038/sdata.2016.18

13 Corlan A. Medline trend: automated yearly statistics of PubMed results for any query. http://dan.corlan.net/medline-trend.html (accessed 1 Jan 2018).

14 Chen H-H, Tsai S-C, Tsai J-H. Mining tables from large scale HTML texts. In: *Proceedings of the 18th conference on Computational linguistics* -. Morristown, NJ, USA: : Association for Computational Linguistics 2000. 166–72. doi:10.3115/990820.990845

15 Penn G, Jianying Hu, Hengbin Luo, *et al*. Flexible Web document analysis for delivery to

narrow-bandwidth devices. In: *Proceedings of Sixth International Conference on Document Analysis and Recognition*. IEEE Comput. Soc 1074–8. doi:10.1109/ICDAR.2001.953951

16 Yoshida M, Yoshida M, Torisawa K. A method to integrate tables of the World Wide Web. In: *IN PROCEEDINGS OF THE INTERNATIONAL WORKSHOP ON WEB DOCUMENT ANALYSIS (WDA 2001*. 2001. 31--34.http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.1832 (accessed 4 Jan 2018).

17 Wang Y, Hu J. A machine learning based approach for table detection on the web. In: *Proceedings of the eleventh international conference on World Wide Web - WWW '02*. New York, New York, USA: : ACM Press 2002. 242. doi:10.1145/511446.511478

18 Son J-W, Lee J-A, Park S-B, *et al*. Discriminating Meaningful Web Tables from Decorative Tables Using a Composite Kernel. In: *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE 2008. 368–71. doi:10.1109/WIIAT.2008.241

19 Wei X, Croft B, Mccallum A. Table Extraction for Answer Retrieval. https://people.cs.umass.edu/~mccallum/papers/TableExtraction-irj06.pdf (accessed 5 Jan 2018).

20 Chavan MM, Shirgave SK. A Methodology for Extracting Head Contents from Meaningful Tables in Web Pages. In: *2011 International Conference on Communication Systems and Network Technologies*. IEEE 2011. 272–7. doi:10.1109/CSNT.2011.66

21 Nagy G. LEARNING THE CHARACTERISTICS OF CRITICAL CELLS FROM WEB TABLES. In: *21st International Conference on Pattern Recognition (ICPR 2012)*.

Tsukuba, Japan: 2012.
http://f4k.dieei.unict.it/proceedings/ICPR2012/media/files/0220.pdf (accessed 5 Jan 2018).

22 Wang J, Wang H, Wang Z, *et al*. Understanding Tables on the Web. In: *Proceedings of the 31st international conference on Conceptual Modeling*. Springer-Verlag 2012. 141–55. doi:10.1007/978-3-642-34002-4_11

23 Hurst MF. *The Interpretation of Tables in Texts*. 2000.https://www.era.lib.ed.ac.uk/handle/1842/7309 (accessed 5 Jan 2018).

24 Pivk A, Cimiano P, Sure Y, *et al*. Transforming arbitrary tables into logical form with TARTAR. *Data Knowl Eng* 2007;**60**:567–95. doi:10.1016/J.DATAK.2006.04.002

25 Wang Y, Phillips IT, Haralick RM. Table structure understanding and its performance evaluation. *Pattern Recognit* 2004;**37**:1479–97. doi:10.1016/J.PATCOG.2004.01.012

26 Liu Y, Bai K, Mitra P, *et al*. TableSeer: automatic table metadata extraction and searching in digital libraries. In: *Proceedings of the 2007 conference on Digital libraries - JCDL '07*. New York, New York, USA: : ACM Press 2007. 91. doi:10.1145/1255175.1255193

27 Wu X, Cao C, Wang Y, *et al*. Extracting Knowledge from Web Tables Based on DOM Tree Similarity. Springer, Cham 2016. 302–13. doi:10.1007/978-3-319-47650-6_24

28 Doush IA, Pontelli E. Non-visual navigation of spreadsheets. *Univers Access Inf Soc* 2013;**12**:143–59. doi:10.1007/s10209-012-0272-1

29 Milosevic N, Gregson C, Hernandez R, *et al*. Disentangling the Structure of Tables in Scientific Literature. Springer, Cham 2016. 162–74. doi:10.1007/978-3-319-41754-7_14

30 Dalvi BB, Cohen WW, Callan J. WebSets: Extracting Sets of Entities from the Web Using Unsupervised Information Extraction. In: *Proceedings of the fifth ACM international*

conference on Web search and data mining - WSDM '12. New York, New York, USA: : ACM Press 2012. 243. doi:10.1145/2124295.2124327

31 Muñoz E, Hogan A, Mileo A. Using linked data to mine RDF from wikipedia's tables. In: *Proceedings of the 7th ACM international conference on Web search and data mining - WSDM '14*. New York, New York, USA: : ACM Press 2014. 533–42. doi:10.1145/2556195.2556266

32 Bizer C, Lehmann J, Kobilarov G, *et al*. DBpedia - A crystallization point for the Web of Data. *Web Semant Sci Serv Agents World Wide Web* 2009;**7**:154–65. doi:10.1016/J.WEBSEM.2009.07.002

33 Wong W, Martinez D, Cavedon L. Extraction of Named Entities from Tables in Gene Mutation Literature. In: *Proceedings of the Workshop on BioNLP*. Boulder, Colorado: 2009. 46–54.http://delivery.acm.org/10.1145/1580000/1572371/p46-wong.pdf?ip=139.52.147.52&id=1572371&acc=OPEN&key=4D4702B0C3E38B35.4D4702B0C3E38B35.4D4702B0C3E38B35.6D218144511F3437&CFID=846931356&CFTOKEN=36503189&__acm__=1515006712_538b25504edffdd96c474e465d94 (accessed 3 Jan 2018).

34 Peng J, Shi X, Sun Y, *et al*. QTLMiner: QTL database curation by mining tables in literature. *Bioinformatics* 2015;**31**:1689–91. doi:10.1093/bioinformatics/btv016

35 Luo D, Peng J, Fu Y. Biotable: A tool to extract semantic structure of table in biology literature. In: *ACM International Conference Proceeding Series*. Association for Computing Machinery 2018. 29–33. doi:10.1145/3309129.3309139

36 Milosevic N, Gregson C, Hernandez R, *et al*. Extracting Patient Data from Tables in Clinical Literature - Case Study on Extraction of BMI, Weight and Number of Patients.

Proc 9th Int Jt Conf Biomed Eng Syst Technol 2016;**5**:223–8. doi:10.5220/0005660102230228

37 Milosevic N, Gregson C, Hernandez R, *et al*. A framework for information extraction from tables in biomedical literature. *Int J Doc Anal Recognit* 2019;**22**:55–78. doi:10.1007/s10032-019-00317-0

38 Shmanina T, Zukerman I, Cheam AL, *et al*. A Corpus of Tables in Full-Text Biomedical Research Publications. In: *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)* . Osaka: 2016. 70–9.http://www.csse.monash.edu.au/research/umnl/data/index.shtml. (accessed 18 Dec 2017).

39 Bateman J, Zock M. *Natural Language Generation*. Oxford University Press 2012. doi:10.1093/oxfordhb/9780199276349.013.0015

40 Reiter E, Dale R. *Building Applied Natural Language Generation Systems*. Cambridge University Press 1997. doi:10.1017/S1351324997001502

41 Barzilay R, Lapata M. Collective content selection for concept-to-text generation. In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*. Morristown, NJ, USA: : Association for Computational Linguistics 2005. 331–8. doi:10.3115/1220575.1220617

42 Zhang X, Lapata M. Chinese Poetry Generation with Recurrent Neural Networks. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg, PA, USA: : Association for Computational Linguistics 2014. 670–80. doi:10.3115/v1/D14-1074

43 Lebret R, Grangier D, Auli M. Neural Text Generation from Structured Data with

Application to the Biography Domain. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: : Association for Computational Linguistics 2016. 1203–13. doi:10.18653/v1/D16-1128

44 Kiddon C, Zettlemoyer L, Choi Y. Globally Coherent Text Generation with Neural Checklist Models. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: : Association for Computational Linguistics 2016. 329–39. doi:10.18653/v1/D16-1032

45 Zang H, Wan X. Towards Automatic Generation of Product Reviews from Aspect-Sentiment Scores. In: *Proceedings of The 10th International Natural Language Generation conference*,. Santiago de Compostela, Spain: 2017. 168–77.http://aclweb.org/anthology/W17-3526 (accessed 19 Dec 2017).

46 Reiter E. Building Natural Language Generation Systems. 2000.

47 Dale R, Geldof S, Prost J-P. CORAL: Using Natural Language Generation for Navigational Assistance. https://pdfs.semanticscholar.org/5ca0/2daaa0d676520d8642ad00351121ff1f29c4.pdf (accessed 19 Dec 2017).

48 Galanis D, Androutsopoulos I. Generating Multilingual Descriptions from Linguistically Annotated OWL Ontologies: the NaturalOWL System. *Proc Elev Eur Work Nat Lang Gener (ENLG 07)* Published Online First: 2007.https://aclanthology.coli.uni-saarland.de/papers/W07-2322/generating-multilingual-descriptions-from-linguistically-annotated-owl-ontologies-the-naturalowl-system (accessed 21 Sep 2017).

49 Reiter, Ehud. An architecture for data-to-text systems. Proc. Elev. Eur. Work. Nat. Lang. Gener. 2007;:97–104.https://dl.acm.org/citation.cfm?id=1610180 (accessed 15 Dec 2017).

50 Portet F, Reiter E, Gatt A, *et al*. Automatic generation of textual summaries from neonatal intensive care data. *Artif Intell* 2009;**173**:789–816. doi:10.1016/J.ARTINT.2008.12.002

51 Mahamood S, Reiter E. Generating affective natural language for parents of neonatal infants. Proc. 13th Eur. Work. Nat. Lang. Gener. 2011;:12–21.https://dl.acm.org/citation.cfm?id=2187685 (accessed 15 Dec 2017).

52 Hunter J, Freer Y, Gatt A, *et al*. BT-Nurse: computer generation of natural language shift summaries from complex heterogeneous medical data. *J Am Med Informatics Assoc* 2011;**18**:621–4. doi:10.1136/amiajnl-2011-000193

53 Hunter J, Freer Y, Gatt A, *et al*. Automatic generation of natural language nursing shift summaries in neonatal intensive care: BT-Nurse. *Artif Intell Med* 2012;**56**:157–72. doi:10.1016/J.ARTMED.2012.09.002

54 Scott D, Hallett C, Fettiplace R. Data-to-text summarisation of patient records: using computer-generated summaries to access patient histories. *Patient Educ Couns* 2013;**92**:153–9. doi:10.1016/j.pec.2013.04.019

55 Yu L, Zhang W, Wang J, *et al*. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. Published Online First: 18 September 2016.http://arxiv.org/abs/1609.05473 (accessed 22 Sep 2017).

56 Hu Z, Yang Z, Liang X, *et al*. Toward Controlled Generation of Text. Published Online First: 2 March 2017.http://arxiv.org/abs/1703.00955 (accessed 13 Sep 2017).

57 Sha L, Mou L, Liu T, *et al*. Order-Planning Neural Text Generation From Structured Data. 2017;:5414–21.http://arxiv.org/abs/1709.00155

58 Liu T, Wang K, Sha L, *et al*. Table-to-text Generation by Structure-aware Seq2seq Learning. Published Online First: 2017.http://arxiv.org/abs/1711.09724

59 Liu T, Luo F, Xia Q, *et al*. Hierarchical Encoder with Auxiliary Supervision for Neural Table-to-Text Generation: Learning Better Representation for Tables. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019. 6786–93. doi:10.1609/aaai.v33i01.33016786

60 Liu T, Luo F, Yang P, *et al*. Towards Comprehensive Description Generation from Factual Attribute-value Tables. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: : Association for Computational Linguistics 2019. 5985–96. doi:10.18653/v1/P19-1600

61 Devlin J, Chang M-W, Lee K, *et al*. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv Prepr arXiv181004805* Published Online First: 10 October 2018.https://arxiv.org/abs/1810.04805 (accessed 12 Oct 2018).

62 Lan Z, Chen M, Goodman S, *et al*. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. Published Online First: 26 September 2019.http://arxiv.org/abs/1909.11942 (accessed 18 Oct 2019).

63 Yang Z, Dai Z, Yang Y, *et al*. XLNet: Generalized Autoregressive Pretraining for Language Understanding. Published Online First: 19 June 2019.http://arxiv.org/abs/1906.08237 (accessed 14 Aug 2019).

64 Vaswani A, Shazeer N, Parmar N, *et al*. Attention Is All You Need. Published Online First: 12 June 2017.http://arxiv.org/abs/1706.03762 (accessed 10 Sep 2018).

65 Radford A, Wu J, Child R, *et al*. Language Models are Unsupervised Multitask Learners. https://github.com/codelucas/newspaper (accessed 25 Feb 2019).

66 Vinyals O, Toshev A, Bengio S, *et al*. Show and Tell: A Neural Image Caption Generator. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

3156–64.https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Vinyals_Show_and_Tell_2015_CVPR_paper.html (accessed 22 Dec 2017).

67 Xu KELVINXU K, Jimmy Lei Ba U, Kiros R, *et al*. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In: *Proceedings of the 32 nd International Conference on Machine Learning*. Lille, France: 2015. http://proceedings.mlr.press/v37/xuc15.pdf (accessed 22 Dec 2017).

68 Kisilev P, Walach E, Barkan E, *et al*. From medical image to automatic medical report generation. *IBM J Res Dev* 2015;**59**:2:1-2:7. doi:10.1147/JRD.2015.2393193

69 Shin H-C, Roberts K, Lu L, *et al*. Learning to Read Chest X-Rays: Recurrent Neural Cascade Model for Automated Image Annotation. *2016 IEEE Conf Comput Vis Pattern Recognit* 2016;:2497–506. doi:10.1109/CVPR.2016.274

70 Hallett C, Power R, Scott D. Summarisation and Visualisation of e-Health Data Repositories. In: *Proceedings of the UK e-science all hands meeting*. 2006. 69–76.http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=4B204794F84D01C1C6D57323B6239848?doi=10.1.1.529.2286&rep=rep1&type=pdf (accessed 6 Oct 2017).

71 Varges S, Bieler H, Stede M, *et al*. SemScribe: Natural Language Generation for Medical Reports. *Proc Eighth Int Conf Lang Resour Eval* Published Online First: 2012.https://aclanthology.coli.uni-saarland.de/papers/L12-1032/l12-1032 (accessed 6 Oct 2017).

72 Guan J, Li R, Yu S, *et al*. Generation of Synthetic Electronic Medical Record Text. *Proc - 2018 IEEE Int Conf Bioinforma Biomed BIBM 2018* 2018;:374–80.http://arxiv.org/abs/1812.02793 (accessed 19 Apr 2020).

73 Liu PJ. Learning to Write Notes in Electronic Health Records. Published Online First: 8 August 2018.http://arxiv.org/abs/1808.02622 (accessed 19 Apr 2020).

74 Lee SH. Natural language generation for electronic health records. *npj Digit Med* 2018;**1**:1–7. doi:10.1038/s41746-018-0070-0

75 Pivovarov R, Elhadad N. Automated methods for the summarization of electronic health records: Table 1. *J Am Med Informatics Assoc* 2015;**22**:938–47. doi:10.1093/jamia/ocv032

76 Hirsch JS, Tanenbaum JS, Lipsky Gorman S, *et al*. HARVEST, a longitudinal patient record summarizer. *J Am Med Informatics Assoc* 2014;**22**:263–74. doi:10.1136/amiajnl-2014-002945

77 Papineni K, Roukos S, Ward T, *et al*. BLEU: a Method for Automatic Evaluation of Machine Translation. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. Philadelphia: 2002. 311–8.http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.19.9416&rep=rep1&type=pdf (accessed 28 Dec 2017).

78 Banerjee S, Lavie A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. 2005. 65–72.https://www.aclweb.org/anthology/W05-0909/ (accessed 3 Oct 2019).

79 Lin C-Y. ROUGE: A Package for Automatic Evaluation of Summaries. In: *Text summarization branches out: Proceedings of the ACL- 04 workshop*. Barcelona, Spain: 2004. 74–81.http://www.aclweb.org/anthology/W04-1013 (accessed 4 Jan 2018).

80 Novikova J, Dušek O, Curry AC, *et al*. Why We Need New Evaluation Metrics for NLG. ;:2241–52.http://aclweb.org/anthology/D17-1238 (accessed 15 Dec 2017).

81 Reiter, Ehud. Task-based evaluation of NLG systems: control vs real-world context. *Proc UCNLG+Eval Lang Gener Eval Work* 2011;:28–32.https://dl.acm.org/citation.cfm?id=2187747 (accessed 15 Dec 2017).

82 Chen DL, Mooney RJ. Learning to Sportscast: A Test of Grounded Language Acquisition. In: *Proceedings of the 25th International Conference on Machine Learning*. Helsinki, Finland: 2008. http://www.cs.utexas.edu/~ml/papers/david-icml-08.pdf (accessed 30 Dec 2017).

83 Dušek O, Novikova J, Rieser V. Evaluating the state-of-the-art of End-to-End Natural Language Generation: The E2E NLG challenge. *Comput Speech Lang* 2020;**59**:123–56. doi:10.1016/J.CSL.2019.06.009

84 Wiseman S, Shieber S, Rush A. Challenges in Data-to-Document Generation. *Proc 2017 Conf Empir Methods Nat Lang Process* 2017;:2253–63.https://aclanthology.coli.uni-saarland.de/papers/D17-1239/d17-1239 (accessed 15 Dec 2017).

85 Moher D, Hopewell S, Schulz KF, *et al*. CONSORT 2010 explanation and elaboration: updated guidelines for reporting parallel group randomised trials. *BMJ* 2010;**340**:c869. doi:10.1136/BMJ.C869

86 Schulz KF, Altman DG, Moher D, *et al*. CONSORT 2010 statement: updated guidelines for reporting parallel group randomised trials. *BMJ* 2010;**340**:c332. doi:10.1136/BMJ.C332

87 PMC Open Access Subset. https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/ (accessed 14 Mar 2020).

88 Soysal E, Wang J, Jiang M, *et al*. CLAMP – a toolkit for efficiently building customized clinical natural language processing pipelines. *J Am Med Informatics Assoc* Published

Online First: 24 November 2017. doi:10.1093/jamia/ocx132

89 NIH Common Data Elements Repository | HealthData.gov. https://healthdata.gov/dataset/nih-common-data-elements-repository (accessed 15 Mar 2020).

90 Nikola Milošević. *A multi-layered approach to information extraction from tables in biomedical documents | Research Explorer | The University of Manchester*. 2018.https://www.research.manchester.ac.uk/portal/en/theses/a-multilayered-approach-to-information-extraction-from-tables-in-biomedical-documents(649fdaee-7754-40a2-aae7-b1bbb88dd87b).html (accessed 11 Jun 2018).

91 Lafferty J, Mccallum A, Pereira FCN, *et al*. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001) http://repository.upenn.edu/cis_papersPublisherURL:http://portal.acm.org/citation.cfm?id=655813PublisherURL:http://portal.acm.org/citation.cfm?id=655813Thisconferencepaperisavailableat ScholarlyCommons:http://repository.upenn.edu/cis_papers/159 (accessed 14 Mar 2020).

92 Zhang Y, Wang J, Tang B, *et al*. UTH_CCB: A Report for SemEval 2014-Task 7 Analysis of Clinical Text. In: *San Diego, California*. Dublin, Ireland: 2014. 802–6.http://www.aclweb.org/anthology/S14-2142 (accessed 11 Dec 2018).

93 Xu J, Zhang Y, Wang J, *et al*. UTH-CCB: The Participation of the SemEval 2015 Challenge-Task 14. In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: 2015. 311–4.http://alt.qcri.org/semeval2015/task14/index.php (accessed 27 Mar 2019).

94 Lee H-J, Xu H, Wang J, *et al*. UTHealth at SemEval-2016 Task 12: an End-to-End System for Temporal Information Extraction from Clinical Notes. *Proc 10th Int Work Semant Eval* 2016;:1292–7. doi:10.18653/v1/S16-1201

95 Xu J, Wu Y, Zhang Y, *et al*. UTH-CCB@BioCreative V CDR Task: Identifying Chemical-induced Disease Relations in Biomedical Text. In: *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*. 2015. 254–9.http://www.chokkan.org/software/crfsuite/ (accessed 12 Mar 2020).

96 Ji Z, Wei Q, Xu H. BERT-based Ranking for Biomedical Entity Normalization. Published Online First: 9 August 2019.http://arxiv.org/abs/1908.03548 (accessed 12 Mar 2020).

97 Wei Q, Ji Z, Si Y, *et al*. Relation Extraction from Clinical Narratives Using Pre-trained Language Models. In: *AMIA Annual Symposium Proceedings*. 2019.

98 Si Y, Wang J, Xu H, *et al*. Enhancing clinical concept extraction with contextual embeddings. *J Am Med Inform Assoc* 2019;**26**:1297–304. doi:10.1093/jamia/ocz096

99 Lee J, Yoon W, Kim S, *et al*. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Published Online First: 25 January 2019.http://arxiv.org/abs/1901.08746 (accessed 7 Mar 2019).

100 Reiter E, Dale R. *Building natural language generation systems*. Cambridge University Press 2000. https://books.google.com/books/about/Building_Natural_Language_Generation_Sys.html?id=qnWQU9C8bDkC (accessed 4 Jan 2018).

101 Junwei Bao*, Duyu Tang, Nan Duan, Zhao Yan, yuanhua Lv, Ming Zhou TZ. Table-to-Text : Describing Table Region with Natural Language. *Aaai* 2018;:1–10.

102 Novikova J, Dušek O, Curry AC, *et al*. Why We Need New Evaluation Metrics for NLG.

Published Online First: 21 July 2017. doi:10.18653/v1/D17-1237

103 Novikova J, Dušek O, Rieser V. RankME: Reliable Human Ratings for Natural Language Generation. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Stroudsburg, PA, USA: : Association for Computational Linguistics 2018. 72–8. doi:10.18653/v1/N18-2012

104 Reiter E. A structured review of the validity of BLEU. Comput. Linguist. 2018;**44**:393–401. doi:10.1162/COLI_a_00322

105 Reiter E. The Architecture of a Natural Language Generation System. In: *Building Natural Language Generation Systems*. 2000. 41–78. doi:10.1017/cbo9780511519857.004

106 Joachims T. Optimizing search engines using clickthrough data. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*. New York, New York, USA: : ACM Press 2002. 133. doi:10.1145/775047.775067

107 Beyond One-Hot: an exploration of categorical variables - Will's Noise. 2015.http://www.willmcginnis.com/2015/11/29/beyond-one-hot-an-exploration-of-categorical-variables/ (accessed 15 Mar 2020).

108 Thorsten J. SVM-rank: Support Vector Machine for Ranking. 2009.http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html (accessed 15 Mar 2020).

109 Reiter E. Microplanning. 2000. 114–58.

110 BATEMAN JA, A. J. Enabling technology for multilingual natural language generation:

the KPML development environment. *Nat Lang Eng* 1997;**3**:S1351324997001514. doi:10.1017/S1351324997001514

111 White M, Rajkumar R. Minimal Dependency Length in Realization Ranking. Association for Computational Linguistics 2012.

112 Gatt A, Reiter E. SimpleNLG: A realisation engine for practical applications. In: *Proceedings of the 12th European Workshop on Natural Language Generation*. Athens, Greece: 2009. 90–3.https://pdfs.semanticscholar.org/4353/a4432e5a030390c9ebcd7f4d4ab867e79290.pdf (accessed 28 Dec 2017).

113 Lexical Systems Group. The SPECIALIST LEXICON. 2018.https://lsg3.nlm.nih.gov/LexSysGroup/Projects/lexicon/current/web/index.html (accessed 7 Jan 2018).

114 Chen Z, Eavani H, Chen W, *et al*. Few-Shot NLG with Pre-Trained Language Model. Published Online First: 20 April 2019.http://arxiv.org/abs/1904.09521 (accessed 26 Oct 2019).

115 Radford A, Wu J, Child R, *et al*. Language Models are Unsupervised Multitask Learners. https://github.com/codelucas/newspaper (accessed 15 Mar 2019).

116 Natural Language Toolkit — NLTK 3.5. https://www.nltk.org/ (accessed 15 Mar 2020).

117 Rouge. https://pypi.org/project/rouge/ (accessed 15 Mar 2020).

118 Shrout PE, Fleiss JL. Intraclass correlations: Uses in assessing rater reliability. *Psychol Bull* 1979;**86**:420–8. doi:10.1037//0033-2909.86.2.420