

본초 및 용량 용어를 이용한 방제구성 자동추출방법에 대한 연구 - 『의종손익』을 중심으로 -

¹대구한의대학교 한의과대학 원전외사학교실 대학원생 · ²동국대학교 한의과대학 원전외사학교실 교수,
³원광대학교 한의과대학 원전학교실 교수 · ⁴대구한의대학교 한의과대학 원전외사학교실 교수
금유정¹, 이병욱², 엄동명³, 송지청^{4*}

A Research on Automatic Data Extract Method for Herbal Formula Combinations Using Herb and Dosage Terminology - Based on 『Euijongsonik』 -

Keum Yujeong¹, Lee Byungwook², Eom Dongmyung³, Song Jichung^{4*}

¹Graduate Student at Dept. of Medical Classics and History, College of Korean
Medicine, Daegu Haany University

²Professor at Dept. of Medical Classics and History, College of Korean Medicine,
Dongguk University

³Professor at Dept. of Medical Classics, College of Korean Medicine, Wonkwang
University

⁴Professor at Dept. of Medical Classics and History, College of Korean Medicine,
Daegu Haany University

Objectives : This research aims to suggest a automatic data extract method for herbal formula combinations from medical classics' texts.

Methods : This research was carried out by using Access of Microsoft Office 365 in Windows 10 of Microsoft. The subject text for extraction was 『Euijongsonik』. Using data sets of herb and dosage terminology, herbal medicinals and their dosages were extracted. Afterwards, using the position value of the character string, the formula combinations were automatically extracted.

Results : The PC environment of this research was Intel Core i7-1065G7 CPU 1.30GHz, with 8GB of RAM and a Windows 10 64bit operation system. Out of 6,115 verses, 19,277 herb-dosage combinations were extracted.

Conclusions : In this research, it was demonstrated that in the case of classical texts that are available as data, knowledge on herbal medicine could be extracted without human or material resources. This suggests an applicability of classical text knowledge to clinical practice.

Key words : formula, herbal formula combination, automatic data extract, 『Euijongsonik』

* Corresponding Author : Song Jichung

Dept. of Medical History, College of Korean Medicine, Daegu Haany University, 1 Haanydae Rd., Kyungsan, Kyungbuk, ROK

Tel : +82-53-819-1826, Fax : +82-53-819-1850, E-mail : prunedias@gamil.com

Received(October 28, 2020), Revised(November 8, 2020), Accepted(November 9, 2020)

Copyright © The Society of Korean Medical Classics. All rights reserved.

Ⓢ This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서론

전 세계적으로 의료정보화 시장이 급격하게 성장함에 따라 우리나라 역시 다양한 방법으로 보건의료 분야 정보화 사업을 추진 중에 있다. 하지만 한의학은 다양한 지식 데이터를 지녔음에도 불구하고 몇몇 문제에 부딪혀 단편적인 형태로의 데이터 축적이나 데이터 검색의 수준에 머물러 있다. 이는 한의학 지식이 대부분 한자어로 구성되어 있어 데이터를 구축하는 것이 쉽지 않기 때문이다. 또한 데이터를 구축하였다 하더라도 데이터 정보처리 및 교환이 어려워 한의학지식을 정보화하는데 어려움이 있으며, 축적하고자 하는 데이터의 대상이 방대하여 필요한 정보를 찾아내는 것도 쉽지 않다.

보건의료에 대한 관심은 항상 존재하였고, 보건의료소비자들의 기대와 욕구는 능동적인 입장에서 의료환경의 변화를 이끌어왔다. 현재 국민들의 건강정보를 알고자 하는 욕구와 안정성이 보장된 양질의 보건의료서비스에 대한 요구 증가는 한의학 분야 또한 한의학 지식의 대중화와 한방 임상 진료의 객관성 및 정확성을 높이기 위한 노력이 필요하다는 점을 시사한다.

특히 현재 한방첩약 국민건강보험 급여화 시범사업이 실시 예정인 가운데, 향후 한방첩약의 건강보험 급여화가 도입되기 위해서는 한방 병·의원의 처방내역을 공개하고 처방내역에 대한 적절성 여부를 판단할 수 있어야 할 것이다. 이와 관련하여 일차적으로는 임상데이터를 활용하는 것이 가장 좋은 방법이라 생각되나, 개인정보 보호에 대한 문제뿐 아니라 특정 질환에 대한 쏠림 현상 및 데이터 제공의 편차에 따른 한계가 있다. 그렇기 때문에 현 상황에서 가장 활용 가능성이 높은 한방 보건의료데이터는 수천년 동안 한의학의 성과가 기록되어 있는 고의서이다.

현재 임상진료에서 사용되는 지식들 역시 고의서에 바탕을 둔 것으로, 특히 고의서에 기록된 방제 지식은 오랜 시간동안 전해 내려오며 검증·축적된 것이다. 『中醫方劑大辭典』¹⁾의 내용만보다라도

96,592개의 처방이 수록되어 있다. 하지만, 방제지식을 임상에 유용하게 활용하기 위해서는 단순히 고의서에서 처방을 검색하는 방법만으로는 부족하다. 처방과 처방 사이의 관계를 분석할 수 있고, 처방 구성에서 각 본초가 가지는 비율과 우선순위를 분석하여 각 구성요소와 병증의 관계를 분석하는 등 입체적인 방법으로 방제 구성을 파악하는 것이 필요하다.

이를 위해서는 우선 대량의 문헌 지식으로부터 많은 인적 자원을 투여하지 않고도 기계적으로 방제 구성을 추출하는 것이 선행되어야 한다. 앞서 『방제학』 교과서에 기재된 방제에 대한 효능 연구²⁾나 본초 구성을 이용한 방제 및 본초의 효능 예측 설정에 관한 연구³⁾ 등 IT기술을 활용하여 임상 적용에 유용함을 돕고자 하는 연구들이 있었다. 그러나 이러한 연구는 특정 서적이거나 특정 대상에 국한해서 진행된 연구이기 때문에 전체 방제 지식을 기계적으로 추출하는 방법론으로는 한계가 있다. 본초 목록을 이용하여 방제의 본초 구성을 자동 추출하는 방법에 대한 연구⁴⁾ 또한 앞서 발표된 바 있지만, 기존 연구에서는 용량 정보를 문자열 그대로 추출하는 것에 그쳤다. 고의서에서는 다양한 단위로 용량을 표기하고 있기 때문에 이러한 용량표기를 하나의 단위로 통일시켜 표현하는 것 또한 임상활용을 유용하게 하는 방법이다.

이에 본 연구에서는 고의서의 원문으로부터 방제 구성을 추출하는 과정에서 용량표기를 모두 그래프[g]으로 변환하여 추출하고, 위치 차의 최솟값(>0)을 이용하여 본초와 용량표기를 조합함으로써 선행연구에서보다 본초-용량표기 조합과정에서의 단계를 단축하는 방법을 제시하고자 한다.

- 2) 박병선, 김은하, 이선아, 이병욱. 『방제학』에 기재된 방제 효능과 본초 구성을 기반으로 도출된 효능의 비교 연구. 대한한의학원전학회지21(1). 2008. pp.79-92.
- 3) 오용택, 김상찬, 이병욱. 본초 구성을 이용한 방제 효능 예측 연구 및 본초의 효능 강도 설정안. 대한한의학원전학회. 대한한의학원전학회지21(2). 2008. pp.49-57.
- 4) 김기욱, 김태열, 이병욱. 본초 목록을 이용한 방제의 본초 구성 자동 추출 방법. 대한한의학원전학회지 27(3). 2014. pp.155-166.

1) 彭懷仁 等 主編, 中醫方劑大辭典. 北京. 人民衛生出版社. 2015.

II. 연구방법

1. 방법

본 연구는 Microsoft社의 Window 10 환경에서 Microsoft Office 365의 Access를 사용해 진행한 것으로 한의서에서 방제구성을 자동 추출하는 것을 목적으로 한다. 방제구성 자동추출 대상 원문은 종합의서인 『醫宗損益』이고, 원문은 한국한의학연구원에서 제공하는 “한의학고전DB”⁵⁾에서 가져왔다. 일차적으로 ‘본초용어’ 데이터 셋⁶⁾과 ‘용량용어’ 데이터 셋을 이용하여 원문에서 본초와 용량을 추출하였으며, 최종적으로 본초와 용량의 관계를 바탕으로 자동적으로 방제구성을 생성하였다.

2. 원리

의서에서 방제구성을 자동 추출하는 원리는 [본초용어]와 [용량용어]를 사용하여 원문에서 방제구성 관련 용어를 추출하고, 문자열의 위치 관계를 이용하여 방제구성을 생성하는 것이다. 이 과정은 3단계로 나뉜다.

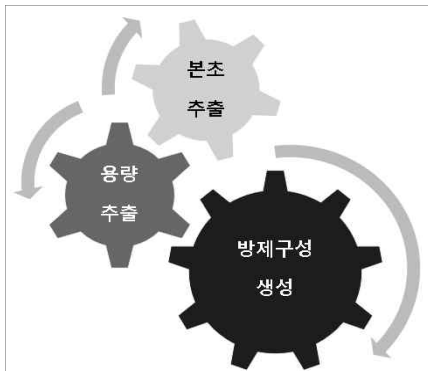


그림 1. 방제구성 자동추출원리

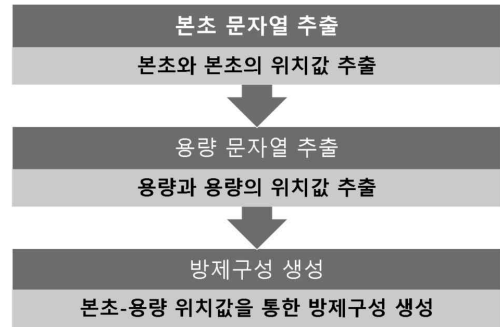


그림 2. 방제구성 자동추출 과정

3. 원리 구현 개념 설명

본 연구의 대상이 된 『의종손익』에서 방제구성을 나타내고 있는 부분을 예로 들어보면 다음(그림 3)과 같다.

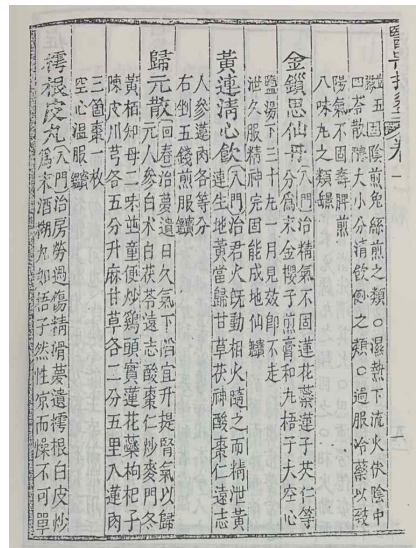


그림 3. 『의종손익』권1·「精」 원문일부

[원문]

[回春] 治夢遺日久氣下陷宜升提腎氣以歸元人蔘白朮白茯苓遠志酸棗仁炒麥門冬黃栢知母二味並童便炒鶴頭實蓮花藥枸杞子陳皮川芎各五分升麻甘草各二分五里入蓮肉三箇棗一枚空心溫服^{醫經}

5) 한의학고전DB: <https://mediclassics.kr/> 2020년 10월 16일 검색

6) 향후 데이터 셋의 명칭은 대괄호를 통해 표시하며, 이는 Access의 테이블명이다.

한국한의학연구원 한의학교전DB에서 해당문장을 가져오면 아래와 같다.

回春治夢遺日久，氣下陷，宜升提腎氣以歸元。

人蔘·白朮·白茯苓·遠志·酸棗仁炒·麥門冬·黃栢·知母二味並童便炒·鷄頭實·蓮花藥·枸杞子·陳皮·川芎各五分，升麻·甘草各二分五厘，入蓮肉三箇，棗一枚，空心溫服。《寶鑑》

이를 통해 방제구성을 자동 추출한 결과는 아래 (그림 4)와 같다.

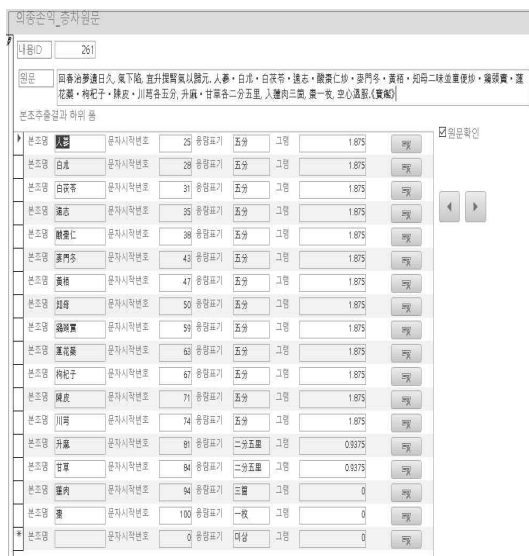


그림 4. [본초용어], [용량용어] 이용 방제구성 자동추출 결과

1) 본초 및 본초문자열 위치정보 추출

[본초용어]7)를 활용하여 본초를 원문에서 찾고, 본초정보와 본초문자열의 첫 글자 문자위치정보8)를 동시에 추출하는 과정이다. 그 결과 『醫宗損益』의

7) 본초용어 테이블은 표제본초명과 이명(異名), 이음동자(異音同字), 간체자(簡體字), 축약어 등을 고려하여 작성하였다. 예를 들어 표제본초명 甘草는 이명으로 國老, 이음동자로 國老(국노)와 國老(국로), 간체자로 国老 등을 link처리하였으며, 熟地黃의 경우 熟苴를 link처리하는 식으로 작성되었다.

8) 문자위치정보는 첫 글자만 나타낸다. 왜냐하면 전체용어의 위치 정보는 첫 글자의 위치정보에 문자열의 길이를 통해 표현할 수 있기 때문이다.

해당원문에서 다음(표 1)과 같은 정보를 추출하였다.

표 1. 본초정보와 본초문자열 위치정보 추출

본초정보	본초위치정보
人蔘	25
白朮	28
白茯苓	31
遠志	35
酸棗仁	38
麥門冬	43
黃栢	47
知母	50
鷄頭實	59
蓮花藥	63
枸杞子	67
陳皮	71
川芎	74
升麻	81
甘草	84
蓮肉	94
棗	100

2) 용량용어 생성

용량용어를 생성하기 위해 ‘一二三四五六七八九十’ 및 ‘壹貳參肆伍陸柒捌玖拾’의 숫자정보와 ‘斤兩錢分厘’ 및 ‘釐’의 무게정보, 숫자정보와 ‘石斗末’ 등 부피정보, 숫자정보와 ‘箇枚粒’ 등 물건을 세는 정보를 각각 조합하여 작성하였다. 또한 각 단위의 무게를 ‘그램’으로 추출할 수 있도록 숫자정보와 무게정보를 연산하여 추출하였다.

다만 연산에 적용하지 않아도 되는 ‘少許’와 ‘等分’은 별도로 테이블에 추가하며, ‘半’은 단위의 50%에 해당이 되므로 문자열 연산으로 추출하면 계산이 되지 않으므로 숫자(문자)와 무게(문자)의 조합에 단위의 50% 그래프를 곱하지 않고 더하는 연산으로 그래프를 추출하여 별도로 테이블에 추가한다.

3) 용량 및 용량문자열 위치정보 추출

[용량용어]를 활용하여 용량을 원문에서 찾고 용량정보와 용량문자열의 첫 글자 문자위치정보를 동시에 추출하는 과정이다. 그 결과 『醫宗損益』의 해당원문에서 다음(표 2)과 같은 정보를 추출하였다.

표 2. 숫자정보와 무게정보 기반 그램추출의 예

숫자 (문자)	아라비아 숫자	X	무게 (문자)	그램	=	용량 단위	그램 추출
一	1		斤	600		一斤	600
二	2		兩	37.5		二斤	1200
三	3		錢	3.75		三斤	1800
...	...		分	0.375		...	
壹	1		里	0.0375			
貳	2		釐	0.0375			

4) 본초와 용량 위치정보 비교

방제구성에서 용량표기는 본초구성 다음에 위치하므로 본초의 위치정보와 용량의 위치정보를 통해 본초와 용량의 관계를 구성할 수 있다. 즉, 용량의 위치정보와 본초의 위치정보 차 중에 그 결과 값이 '>0'인 것만 정리하면 본초와 용량의 관계를 규정할 수 있다.

용량 위치정보에서 본초 위치정보를 빼고 그 결과 값이 '>0'인 것을 정리하면 다음(표 3)과 같다.

표 3. 용량정보, 용량문자열 위치정보, 그램 추출

용량정보	용량위치정보	그램
五分	77	1.875
二分五里	87	0.9375
三箇	96	
一枚	101	

이처럼 본초위치정보와 용량위치정보의 차 중 결과 값이 '>0'인 것은 다수 존재하는데 결과 값 중 최솟값을 이용하면 본초와 용량의 관계를 규정할 수 있다. 표 4에서 최솟값은 굵은 글씨와 함께 밑줄로 표시하였다.

이를 바탕으로 본초와 용량의 관계를 바탕으로 처방구성을 나타내면 다음(표 4)과 같다.

표 4. 용량위치정보 본초위치정보 차(결과값>0)

본초정보	본초 위치정보	본초위치-용량위치 결과값(>0)			
		五分 77	二分五里 87	三箇 96	一枚 101
人蔘	25	52	62	71	76
白朮	28	49	59	68	73
白茯苓	31	46	56	65	70
遠志	35	42	52	61	66
酸棗仁	38	39	49	58	63
麥門冬	43	34	44	53	58
黃栢	47	30	40	49	54
知母	50	27	37	46	51
鷄頭實	59	18	28	37	42
蓮花藥	63	14	24	33	38
枸杞子	67	10	20	29	34
陳皮	71	6	16	25	30
川芎	74	3	13	22	27
升麻	81		6	15	20
甘草	84		3	12	17
蓮肉	94			2	7
棗	100				1

5. 처방구성 추출

본초정보	용량정보	그램
人蔘	五分	1.875
白朮	五分	1.875
白茯苓	五分	1.875
遠志	五分	1.875
酸棗仁	五分	1.875
麥門冬	五分	1.875
黃栢	五分	1.875
知母	五分	1.875
鷄頭實	五分	1.875
蓮花藥	五分	1.875
枸杞子	五分	1.875
陳皮	五分	1.875
川芎	五分	1.875
升麻	二分五里	0.9375
甘草	二分五里	0.9375
蓮肉	三箇	0
棗	一枚	0

Ⅲ. 본문

본 연구에서는 다음과 같이 데이터베이스를 구성하고 Macro를 작성하여 방제 구성 자동추출과정을 완성하였다.

1. 방제구성 자동추출 데이터베이스의 구성

연구방법에서 기술한 내용을 구현하기 위해 다음과 같은 데이터베이스를 작성해야 한다.

1) 테이블 구성9)

(1) 숫자 테이블: [숫자]

용량을 계산하기 위한 숫자가 저장되는 테이블을 작성해야 하며 숫자, 아라비아숫자 필드를 갖는다.

(2) 단위 테이블: [단위]

용량을 계산하기 위한 단위가 저장되는 테이블을 작성해야 하며 단위, 그램 필드를 갖는다.

(3) 텍스트 테이블: [의종손익_층차원문]

원문 저장을 위한 테이블을 작성해야 하며 내용 ID, 권수명, 문명, 소재목, 처방명, 원문, 확인 필드를 갖는다.

(4) 문장분할 테이블: [문장분할]

Access 프로그램의 가독성을 위해 255글자 이내의 원문으로 분할하는 테이블을 작성해야 하며 내용 ID, 문장분할, 순서, 문자시작번호 필드를 갖는다.

(5) 본초용어 테이블: [용어목록]

본초정보를 추출하는데 사용되는 본초용어 목록 저장을 위한 테이블을 작성해야 하며 (본초)표준용어, (본초)비표준용어 필드를 갖는다.

(6) 용량용어 테이블: [용량표기]

용량정보를 추출하는데 사용되는 용량용어 목록

저장을 위한 테이블을 작성해야 하며 숫자, 단위, 용량단위, 그램 필드를 갖는다.

(7) 본초길이 최대값 테이블: [본초길이MAX]

본초길이 중 최대값을 구하는 테이블로 본초길이 최대값 필드를 갖는다.

(8) 최대값 본초 추출 테이블: [길이MAX본초추출]

최대길이의 본초만 선택해 구성하는 테이블로 내용ID, 순서, 본초명, 용량단위, 본초길이, 문자시작번호, 필드를 갖는다.

(9) 본초 저장 테이블: [본초추출결과]

원문에서 추출한 본초를 저장하고, 최종적으로 본초-용량의 정보를 통해 방제구성 정보를 저장하는 테이블로 내용ID, 순서, 본초명, 용량단위, 본초길이, 문자시작번호, 그램 필드를 갖는다.

(10) 용량 저장 테이블: [임시저장]

원문에서 1차적으로 추출한 용량을 임시저장하는 테이블로 내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램 필드를 갖는다.

(11) 2회 이상 출현하는 용량 저장 테이블: [임시저장_0]

원문에서 1회 이상 출현하는 용량정보를 저장하는 테이블로 내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램 필드를 갖는다.

(12) 2개 조합 용량 저장 테이블: [임시저장_4단위]

원문에서 추출한 용량이 연속적으로 표기되어 있는 용량을 추출하는 테이블로 내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램 필드를 갖는다.

(13) 3개 조합 용량 저장 테이블: [임시저장_6단위]

9) 실제 Access에서 사용한 테이블 명칭을 괄호 제목 뒤에 병기하였다.

원문에서 추출한 용량이 3연속으로 표기되어 있는 용량을 추출하는 테이블로 내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램 필드를 갖는다.

(14) 용량길이 최대값 테이블: [본초용량길이 MAX]

용량길이 중 최대값을 구하는 테이블로 본초용량길이최대값 필드를 갖는다.

(15) 용량 저장 테이블: [임시본초용량추출결과]

1차로 추출한 용량, 2회 이상 출현하는 용량, 연속적으로 표기되어 있는 용량, 3연속으로 표기되어 있는 용량을 모두 임시저장하는 테이블로 내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램 필드를 갖는다.

(16) 최대값 용량 추출 테이블: [길이MAX본초용량추출]

최대길이의 용량만 선택해 구성하는 테이블로 내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램 필드를 갖는다.

(17) 본초위치 및 용량위치 차이 저장 테이블: [본초및용량단위위치차]

본초 및 용량의 위치차 계산값을 저장하는 테이블로 내용ID, 순서, 본초명, 용량단위, 문자시작번호, 필드를 갖는다.

(18) 본초위치 및 용량위치 차이 최소값 저장 테이블: [위치차_최소값]

본초 및 용량의 위치차 계산 후 최소값을 저장하는 테이블로 내용ID, 순서, 본초명, 본초_용량위치차의최소값, 그램 필드를 갖는다.

2) 쿼리 구성

본 연구에서 방제구성을 자동적으로 추출하는 과정은 삭제, 추가, 업데이트 쿼리를 생성한 뒤 과정 순서에 따라 배열하고 이를 실행하는 방식으로 구성하였다. 쿼리 내용은 다음과 같다.

(1) 삭제 쿼리

① 07용량표기삭제

[용량표기] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 용량표기.* FROM 용량표기;

② 011본초추출결과삭제

[본초추출결과] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 본초추출결과.* FROM 본초추출결과;

③ 012임시저장삭제

[임시저장] 테이블 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 임시저장.* FROM 임시저장;

④ 012_1임시저장_0삭제

[임시저장_0] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 임시저장_0.* FROM 임시저장_0;

⑤ 013임시저장_4단위삭제

[임시저장_4단위] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 임시저장_4단위.* FROM 임시저장_4단위;

⑥ 014임시저장_6단위삭제

[임시저장_6단위] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 임시저장_6단위.* FROM 임시저장_6단위;

⑦ 0141임시본초용량추출결과삭제

[임시본초용량추출결과] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 임시본초용량추출결과.* FROM 임시본초용량추출결과;

⑧ 0151본초용량길이MAX삭제

[본초용량길이MAX] 내용을 삭제하는 쿼리로 쿼

리문장은 아래와 같다.

DELETE 본초용량길이MAX.* FROM 본초용량길이MAX;

⑨ 0152길이MAX본초용량추출삭제

[길이MAX본초용량추출] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 길이MAX본초용량추출.* FROM 길이MAX본초용량추출;

⑩ 0153임시본초용량추출결과_삭제결과 정리

[임시본초용량추출결과]에 표기된 '삭제' 표기 Data를 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 임시본초용량추출결과.용량단위 FROM 임시본초용량추출결과 WHERE (((임시본초용량추출결과.용량단위)="삭제"));

⑪ 021본초길이MAX삭제

[본초길이MAX] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 본초길이MAX.* FROM 본초길이MAX;

⑫ 022길이MAX본초추출삭제

[길이MAX본초추출] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 길이MAX본초추출.* FROM 길이MAX본초추출;

⑬ 023본초추출결과_삭제정리

[본초추출결과]에 표기된 '삭제' 표기 Data를 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 본초추출결과.본초명 FROM 본초추출결과 WHERE (((본초추출결과.본초명)="삭제"));

⑭ 031본초용량단위위치차삭제

[본초용량단위위치차] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 본초및용량단위위치차.* FROM 본초및용량단위위치차;

⑮ 032위치차_최소값삭제

[위치차_최소값] 내용을 삭제하는 쿼리로 쿼리문장은 아래와 같다.

DELETE 위치차_최소값.* FROM 위치차_최소값;

(2) 업데이트 쿼리

① 쿼리23

[길이MAX본초추출]과 [본초추출결과]를 활용하여 동일 '내용ID'와 '순서'를 연결한 후 [길이MAX본초추출]의 '본초길이'보다는 짧으면서 [길이MAX본초추출]의 위치가 [본초추출결과]의 본초 시작위치 및 끝 위치 내에 있는 본초에 '삭제'를 표시하기 위한 쿼리이며, 쿼리문장은 아래와 같다.

UPDATE 길이MAX본초추출 INNER JOIN 본초추출결과 ON (길이MAX본초추출.내용ID = 본초추출결과.내용ID) AND (길이MAX본초추출.순서 = 본초추출결과.순서) SET 본초추출결과.본초명 = "삭제" WHERE (((본초추출결과.본초길이)<길이MAX본초추출.본초길이) And ((본초추출결과.문자시작번호)>=(길이MAX본초추출.문자시작번호) And (본초추출결과.문자시작번호)<=(길이MAX본초추출.문자시작번호+ 길이MAX본초추출.본초길이))));

② 쿼리24

[본초길이MAX]를 여러번 사용하여 길이값에 따른 중복되는 본초를 삭제하기 위해 사용되는 것으로 최대길이를 이용해 사용한 후, 다음 단계에서 값을 '-1' 단위로 바꾸는 쿼리이며, 쿼리문장은 아래와 같다.

UPDATE 본초길이MAX SET 본초길이MAX.본초길이의최대값 = [본초길이MAX].[본초길이의최대값]-1 WHERE (((본초길이MAX.본초길이의최대값)>0));

③ 쿼리33

[본초추출결과], [본초및용량단위위치차], [위치차_최소값]을 이용해 동일 '내용ID', '순서', '본초명'을 연결하고 '본초-용량 위치차' 중 최소값을 구하는 쿼리이며, 쿼리문장은 아래와 같다.

UPDATE 본초추출결과 INNER JOIN (본초및용량단위위치차 INNER JOIN 위치차_최소값 ON (본초및용량단위위치차.내용ID = 위치차_최소값.내용ID) AND (본초및용량단위위치차.순서 = 위치차_최소값.순서) AND (본초및용량단위위치차.본초명 = 위치차_최소값.본초명) AND (본초및용량단위위치차.본초_용량위치차 = 위치차_최소값.본초_용량위치차의최소값)) ON (본초추출결과.본초명 = 본초및용량단위위치차.본초명) AND (본초추출결과.순서 = 본초및용량단위위치차.순서) AND (본초추출결과.내용ID = 본초및용량단위위치차.내용ID) SET 본초추출결과.용량표기 = [본초및용량단위위치차].[용량단위], 본초추출결과.그램 = [본초및용량단위위치차].[그램];

④ 쿼리153

[길이]MAX본초용량추출]과 [임시본초용량추출결과]를 이용해서 동일 '내용ID'와 '순서'를 연결한 후 [길이]MAX본초용량추출]의 '용량길이'보다는 짧으면서 [길이]MAX본초용량추출]의 위치가 [임시본초용량추출결과]의 용량 시작위치 및 끝 위치 내에 있는 용량에 '삭제'를 표시하기 위한 쿼리이며, 쿼리문장은 아래와 같다.

UPDATE 길이MAX본초용량추출 INNER JOIN 임시본초용량추출결과 ON (길이MAX본초용량추출.순서 = 임시본초용량추출결과.순서) AND (길이MAX본초용량추출.내용ID = 임시본초용량추출결과.내용ID) SET 임시본초용량추출결과.용량단위 = "삭제"

WHERE (((임시본초용량추출결과.용량길이)<[길이]MAX본초용량추출].[용량길이]) AND ((임시본초용량추출결과.용량단위위치)>=[길이]MAX본초용량추출].[용량단위위치] And (임시본초용량추출결과.용량단위위치)<=[길이]MAX본초용량추출].[용량단위위치]+ [길이]MAX본초용량추출].[용량길이]));

⑤ 쿼리154

[본초용량길이MAX]를 여러번 사용하여 길이값에 따른 중복되는 용량을 삭제하기 위해 사용되는 것으로 최대길이를 이용해 사용 후, 다음 단계에서

값을 '-1' 단위로 바꾸는 쿼리이며, 쿼리문장은 아래와 같다.

UPDATE 본초용량길이MAX SET 본초용량길이MAX.본초용량길이의최대값 = [본초용량길이MAX].[본초용량길이의최대값]-1 WHERE (((본초용량길이MAX.본초용량길이의최대값)>0));

(3) 추가 쿼리

① 쿼리1

[의중손익_첨차원문]의 '원문' 필드를 Access가 데이터처리를 할 수 있도록 255글자 이내¹⁰⁾로 문장을 분할하고, 원문장의 길이를 파악할 수 있도록 분할 다음 문장은 분할 문장 앞의 문장 길이 '250'을 표기하여 [문장분할]에 추가하는 쿼리로 [문장분할]의 내용ID, 문장분할, 문자시작번호 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 문장분할 (내용ID, 문장분할, 문자시작번호)

SELECT 의중손익_첨차원문.내용ID, Mid([의중손익_첨차원문].[원문],1,250) AS 문장분할, 0 AS 문자시작번호

FROM 의중손익_첨차원문

WHERE (((Mid([의중손익_첨차원문].[원문],1,250)) Is Not Null))

ORDER BY 의중손익_첨차원문.내용ID;

② 쿼리2, 3, 4, 5, 6

문장의 길이에 따라 문장분할을 250, 500, 750, 1000, 1250 등 250글자 단위로 분할하는 과정으로 쿼리1의 과정을 반복하지만 문장의 원길이를 표기하여 [문장분할]의 내용ID, 문장분할, 문자시작번호 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 문장분할 (내용ID, 문장분할, 문자시작번호)

SELECT 의중손익_첨차원문.내용ID, Mid([의중손익_첨차원문].[원문],251¹¹⁾,250) AS 문장분할,

10) 본 연구에서는 250글자로 분할하였다.

11) 쿼리3은 501, 쿼리4는 751, 쿼리5는 1001, 쿼리 6은 1251로 작성되며, 『醫宗損益』의 문장은 1500자가 넘는

250¹²⁾ AS 문자시작번호
 FROM 의중손익_층차원문
 WHERE (((Len([의중손익_층차원문].[원문]))>250¹³⁾))
 ORDER BY 의중손익_층차원문.내용ID;

③ 쿼리7

[숫자]와 [단위]를 이용하여 용량을 연산하는 쿼리로 [용량표기]의 숫자, 단위, 용량단위, 그램 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 용량표기 (숫자, 단위, 용량단위, 그램)
 SELECT distinct 숫자.숫자, 단위.단위, [숫자].[숫자] & [단위].[단위] AS 용량단위, [숫자].[아라비아]*[단위].[그램] AS 그램
 FROM 단위, 숫자;

④ 쿼리11

[용어목록]을 이용하여 [문장분할]에서 [용어목록]과 일치하는 본초용어를 추출하여 [본초추출결과]에 추가하는 쿼리로 [본초추출결과]의 내용ID, 순서, 본초명, 본초길이, 문자시작번호, 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 본초추출결과 (내용ID, 순서, 본초명, 본초길이, 문자시작번호)
 SELECT DISTINCT 문장분할.내용ID, 문장분할.순서, 용어목록.비표준용어, Len([용어목록].[비표준용어]) AS 본초길이, InStr(1,[문장분할].[문장분할],[용어목록].[비표준용어])+ [문장분할].[문자시작번호] AS 문자시작번호
 FROM 문장분할 INNER JOIN 용어목록 ON 문장분할.문장분할 like"*"& 용어목록.비표준용어 &"*"

FROM 문장분할 INNER JOIN 용어목록 ON 문장분할.문장분할 like"*"& 용어목록.비표준용어 &"*"

내용이 없었기에 쿼리6까지 작성되었다.
 12) 쿼리3은 500, 쿼리4는 750, 쿼리5는 1000, 쿼리 6은 1250로 작성되며, 『醫宗損益』의 문장은 1500자가 넘는 내용이 없었기에 쿼리6까지 작성되었다.
 13) 쿼리3은 500, 쿼리4는 750, 쿼리5는 1000, 쿼리 6은 1250로 작성되며, 『醫宗損益』의 문장은 1500자가 넘는 내용이 없었기에 쿼리6까지 작성되었다.

WHERE ((([용어목록.구분]="본초"))
 ORDER BY 문장분할.내용ID, 문장분할.순서, InStr(1,[문장분할].[문장분할],[용어목록].[비표준용어])+ [문장분할].[문자시작번호];

⑤ 쿼리12

[용량표기]를 이용하여 [문장분할]에서 [용량표기]와 일치하는 용량용어를 추출하여 [임시저장]에 추가하는 쿼리로 [임시저장]의 내용ID, 순서, 용량단위, 용량길이, 용량단위위치 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 임시저장 (내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램)
 SELECT DISTINCT 문장분할.내용ID, 문장분할.순서, 용량표기.용량단위, Len([용량표기].[용량단위]) AS 용량길이, InStr(1,[문장분할].[문장분할],[용량표기].[용량단위])+ [문장분할].[문자시작번호] AS 용량단위위치, 용량표기.그램

FROM 문장분할 INNER JOIN 용량표기 ON 문장분할.문장분할 like"*"& 용량표기.용량단위&"*"

ORDER BY 문장분할.내용ID, 문장분할.순서, InStr(1,[문장분할].[문장분할],[용량표기].[용량단위])+ [문장분할].[문자시작번호];

⑥ 쿼리12_1

[용량표기]를 이용하여 [문장분할]에서 [용량표기]와 일치하는 용량용어를 추출할 때 [임시저장]에서 추출된 '용량' 데이터에 국한해 그 뒤에 나오는 동일 용량을 추출해 [임시저장_0]에 추가하는 쿼리로 [임시저장_0]의 내용ID, 순서, 용량단위, 용량길이, 용량단위위치 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 임시저장_0 (내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램)

SELECT DISTINCT 문장분할.내용ID, 문장분할.순서, 용량표기.용량단위, Len([용량표기].[용량단위]) AS 용량길이, InStr([임시저장].[용량단위위치])+ 1,[문장분할].[문장분할],[용량표기].[용량단위])+ [문장분할].[문자시작번호] AS 용량단위위치,

용량표기.그램

FROM 문장분할 INNER JOIN (용량표기 INNER JOIN 임시저장 ON 용량표기.용량단위 = 임시저장.용량단위) ON (문장분할.문장분할 like"*"& 용량표기.용량단위&"*") AND (문장분할.순서 = 임시저장.순서) AND (문장분할.내용ID = 임시저장.내용ID)

WHERE (((InStr([임시저장].[용량단위위치]+1, [문장분할].[문장분할],[용량표기].[용량단위])+ [문장분할].[문자시작번호])>"0"))

ORDER BY 문장분할.내용ID, 문장분할.순서, InStr([임시저장].[용량단위위치]+1,[문장분할].[문장분할],[용량표기].[용량단위])+ [문장분할].[문자시작번호];

⑦ 쿼리12_2

[임시저장_0] 내용 중 중복되는 것을 제외하고 [임시저장]에 추가하는 쿼리로 [임시저장]의 내용ID, 순서, 용량단위, 용량길이, 용량단위위치 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 임시저장 (내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램)

SELECT DISTINCT 임시저장_0.내용ID, 임시저장_0.순서, 임시저장_0.용량단위, 임시저장_0.용량길이, 임시저장_0.용량단위위치, 임시저장_0.그램

FROM 임시저장 INNER JOIN 임시저장_0 ON (임시저장.순서 = 임시저장_0.순서) AND (임시저장.내용ID = 임시저장_0.내용ID)

WHERE (((임시저장_0.용량단위)>[임시저장].[용량단위]));

⑧ 쿼리13

[임시저장] 2개를 활용하여 연속되는 용량단위를 조합해 [임시저장_4단위]로 추가하는 쿼리이며, [임시저장_4단위]의 내용ID, 순서, 용량단위, 용량길이, 용량단위위치 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 임시저장_4단위 (내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램)

SELECT 임시저장.내용ID, 임시저장.순서, [임시

저장].[용량단위] & [임시저장_1].[용량단위] AS 용량단위, [임시저장].[용량길이]+ [임시저장_1].[용량길이] AS 용량길이, 임시저장.용량단위위치, [임시저장].[그램]+ [임시저장_1].[그램] AS 그램

FROM 임시저장 INNER JOIN 임시저장 AS 임시저장_1 ON (임시저장.순서 = 임시저장_1.순서) AND (임시저장.내용ID = 임시저장_1.내용ID)

WHERE (((임시저장_1.용량단위위치)=[임시저장].[용량길이]+ [임시저장].[용량단위위치]));

⑨ 쿼리14

[임시저장]과 [임시저장_4단위]를 활용하여 연속되는 용량단위를 조합해 [임시저장_6단위]로 추가하는 쿼리이며, [임시저장_6단위]의 내용ID, 순서, 용량단위, 용량길이, 용량단위위치 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 임시저장_6단위 (내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램)

SELECT 임시저장.내용ID, 임시저장.순서, [임시저장].[용량단위] & [임시저장_4단위].[용량단위] AS 용량단위, [임시저장].[용량길이]+ [임시저장_4단위].[용량길이] AS 용량길이, 임시저장.용량단위위치, [임시저장].[그램]+ [임시저장_4단위].[그램] AS 그램

FROM 임시저장 INNER JOIN 임시저장_4단위 ON (임시저장.내용ID = 임시저장_4단위.내용ID) AND (임시저장.순서 = 임시저장_4단위.순서)

WHERE (((임시저장_4단위.용량단위위치)=[임시저장].[용량길이]+ [임시저장].[용량단위위치]))

ORDER BY 임시저장.내용ID, 임시저장.순서;

⑩ 쿼리21

[본초추출결과]의 본초길이 최대값을 [본초길이MAX]에 추가하는 쿼리이며, [본초길이MAX]의 본초길이 최대값 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 본초길이MAX (본초길이의최대값)

SELECT Max(본초추출결과.본초길이) AS 본초길이의최대값

FROM 본초추출결과;

⑪ 쿼리22

[본초길이MAX]에 추출된 최대길이 본초를 [본초추출결과]에서 선택한 후 [길이MAX본초추출]에 추가하는 쿼리이며, [길이MAX본초추출]의 내용ID, 순서, 본초명, 용량단위, 본초길이, 문자시작번호 필드에 추가된다. 쿼리문장은 아래와 같다.

```
INSERT INTO 길이MAX본초추출 (내용ID, 순서, 본초명, 용량단위, 본초길이, 문자시작번호, 그램 )
```

```
SELECT DISTINCT 본초추출결과.내용ID, 본초추출결과.순서, 본초추출결과.본초명, 본초추출결과.용량단위, 본초추출결과.본초길이, 본초추출결과.문자시작번호, 본초추출결과.그램
```

```
FROM 본초길이MAX INNER JOIN 본초추출결과 ON 본초길이MAX.본초길이의최대값 = 본초추출결과.본초길이;
```

⑫ 쿼리31

[본초추출결과]와 [임시본초용량추출결과]를 바탕으로 본초 및 용량의 위치차 계산후 최소값을 [본초및용량단위위치차]로 추가하는 쿼리이며, [본초및용량단위위치차]의 내용ID, 순서, 본초명, 용량단위, 그램, 문자시작번호, 본초_용량위치차 필드에 추가된다. 쿼리문장은 아래와 같다.

```
INSERT INTO 본초및용량단위위치차 (내용ID, 순서, 본초명, 용량단위, 그램, 문자시작번호, 본초_용량위치차 )
```

```
SELECT 본초추출결과.내용ID, 본초추출결과.순서, 본초추출결과.본초명, 임시본초용량추출결과.용량단위 AS 용량단위, 임시본초용량추출결과.그램, 본초추출결과.문자시작번호, Min([임시본초용량추출결과].[용량단위위치]-[본초추출결과].[문자시작번호]) AS 본초_용량위치차
```

```
FROM 본초추출결과 INNER JOIN 임시본초용량추출결과 ON (본초추출결과.순서 = 임시본초용량추출결과.순서) AND (본초추출결과.내용ID = 임시본초용량추출결과.내용ID)
```

```
GROUP BY 본초추출결과.내용ID, 본초추출결과.
```

순서, 본초추출결과.본초명, 임시본초용량추출결과.용량단위, 임시본초용량추출결과.그램, 본초추출결과.문자시작번호

```
HAVING (((Min([임시본초용량추출결과].[용량단위위치]-[본초추출결과].[문자시작번호]))>0))
```

```
ORDER BY 본초추출결과.내용ID, 본초추출결과.순서, 본초추출결과.문자시작번호, Min([임시본초용량추출결과].[용량단위위치]-[본초추출결과].[문자시작번호]);
```

⑬ 쿼리32

[본초및용량단위위치차]테이블 2개로 위치차가 제일 작은 것을 [위치차_최소값]에 추가하는 쿼리이며, [위치차_최소값]의 내용ID, 순서, 본초명, 본초_용량위치차의최소값 필드에 추가된다. 쿼리문장은 아래와 같다.

```
INSERT INTO 위치차_최소값 (내용ID, 순서, 본초명, 본초_용량위치차의최소값)
```

```
SELECT 본초및용량단위위치차.내용ID, 본초및용량단위위치차.순서, 본초및용량단위위치차.본초명, Min(본초및용량단위위치차.본초_용량위치차) AS 본초_용량위치차의최소값
```

```
FROM 본초및용량단위위치차 INNER JOIN 본초및용량단위위치차 AS 본초및용량단위위치차_1 ON (본초및용량단위위치차.본초명 = 본초및용량단위위치차_1.본초명) AND (본초및용량단위위치차.순서 = 본초및용량단위위치차_1.순서) AND (본초및용량단위위치차.내용ID = 본초및용량단위위치차_1.내용ID)
```

```
GROUP BY 본초및용량단위위치차.내용ID, 본초및용량단위위치차.순서, 본초및용량단위위치차.본초명;
```

⑭ 쿼리141

[임시저장] 내용을 [임시본초용량추출결과]로 추가하는 쿼리이며, [임시본초용량추출결과]의 내용ID, 순서, 용량단위, 용량길이, 용량단위위치 필드에 추가된다. 쿼리문장은 아래와 같다.

```
INSERT INTO 임시본초용량추출결과 (내용ID, 순서, 용량단위, 용량길이, 용량단위위치, 그램 )
```

```
SELECT 임시저장.내용ID, 임시저장.순서, 임시
```

저장.용량단위, 임시저장.용량길이, 임시저장.용량단
 위위치, 임시저장.그램

FROM 임시저장

GROUP BY 임시저장.내용ID, 임시저장.순서, 임
 시저장.용량단위, 임시저장.용량길이, 임시저장.용량
 단위위치, 임시저장.그램;

⑮ 쿼리142

[임시저장_4단위] 내용을 [임시본초용량추출결
 과]로 추가하는 쿼리이며, [임시본초용량추출결과]
 의 내용ID, 순서, 용량단위, 용량길이, 용량단위위치
 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 임시본초용량추출결과 (내용ID,
 순서, 용량단위, 용량길이, 용량단위위치, 그램)

SELECT 임시저장_4단위.내용ID, 임시저장_4단
 위.순서, 임시저장_4단위.용량단위, 임시저장_4단위.
 용량길이, 임시저장_4단위.용량단위위치, 임시저장_4
 단위.그램

FROM 임시저장_4단위;

⑯ 쿼리143

[임시저장_6단위] 내용을 [임시본초용량추출결
 과]로 추가하는 쿼리이며, [임시본초용량추출결과]
 의 내용ID, 순서, 용량단위, 용량길이, 용량단위위치
 필드에 추가된다. 쿼리문장은 아래와 같다.

INSERT INTO 임시본초용량추출결과 (내용ID,
 순서, 용량단위, 용량길이, 용량단위위치, 그램)

SELECT 임시저장_6단위.내용ID, 임시저장_6단
 위.순서, 임시저장_6단위.용량단위, 임시저장_6단위.
 용량길이, 임시저장_6단위.용량단위위치, 임시저장_6
 단위.그램

FROM 임시저장_6단위;

⑰ 쿼리151

본초용량길이최대값을 [본초용량길이MAX] 테이
 블에 추가하는 쿼리이며, [본초용량길이MAX]의 용
 량길이의최대값 필드에 추가된다. 쿼리문장은 아래
 와 같다.

INSERT INTO 본초용량길이MAX (용량길이의

최대값)

SELECT Max(임시본초용량추출결과.용량길이)
 AS 용량길이의최대값

FROM 임시본초용량추출결과;

⑱ 쿼리152

[임시본초용량추출결과] 중 최대길이 본초용량만
 추출해 [길이MAX본초용량추출]에 추가하는 쿼리이
 며, [길이MAX본초용량추출]의 내용ID, 순서, 용량
 단위, 용량길이, 용량단위위치 필드에 추가된다. 쿼
 리문장은 아래와 같다.

INSERT INTO 길이MAX본초용량추출 (내용ID,
 순서, 용량단위, 용량길이, 용량단위위치, 그램)

SELECT 임시본초용량추출결과.내용ID, 임시본초
 용량추출결과.순서, 임시본초용량추출결과.용량단위,
 임시본초용량추출결과.용량길이, 임시본초용량추출
 결과.용량단위위치, 임시본초용량추출결과.그램

FROM 본초용량길이MAX INNER JOIN 임시본초
 용량추출결과 ON 본초용량길이MAX.용량길이의최
 대값 = 임시본초용량추출결과.용량길이;

2. 방제구성 자동추출 과정

Access에 작성한 쿼리를 순차적으로 진행하기
 위해 매크로를 작성하였고, 그 순서는 다음과 같다.

1) Macro1: 문장분할 단계

- (1) [쿼리1] 실행
- (2) [쿼리2] 실행
- (3) [쿼리3] 실행
- (4) [쿼리4] 실행
- (5) [쿼리5] 실행
- (6) [쿼리6] 실행

2) Macro2: 본초용어추출 단계

2-1) 본초문자열최대값추출 단계

- 1) [011본초추출결과삭제] 실행
- 2) [쿼리11] 실행
- 3) [021본초길이MAX삭제] 실행
- 4) [쿼리21] 실행

- 2-2) 중복본초삭제 단계(반복10회)
- (1) [022길이MAX본초추출삭제] 실행
 - (2) [쿼리22] 실행
 - (3) [쿼리23] 실행
 - (4) [023본초추출결과_삭제정리] 실행
 - (5) [쿼리24] 실행

3) Macro3: 용량용어추출 단계

3-1) 용량용어추출 1단계

- (1) [07용량단위삭제] 실행
- (2) [쿼리7] 실행
- (3) [012임시저장삭제] 실행
- (4) [쿼리012] 실행

3-2) 용량용어추출 2단계

- (1) [012_1임시저장_0삭제] 실행
- (2) [쿼리12_1] 실행
- (3) [쿼리12_2] 실행

3-3) 용량용어추출 3단계

- (1) [013임시저장_4단위삭제] 실행
- (2) [쿼리13] 실행
- (3) [014임시저장_6단위삭제] 실행
- (4) [쿼리14] 실행
- (5) [0141임시본초용량추출결과삭제] 실행
- (6) [쿼리141] 실행
- (7) [쿼리142] 실행
- (8) [쿼리143] 실행

3-4) 용량용어추출정리 단계

- (1) [0151본초용량길이MAX삭제] 실행
- (2) [쿼리151] 실행

3-5) 용량용어정제 단계(10회)

- (1) [0152길이MAX본초용량추출삭제] 실행
- (2) [쿼리152] 실행
- (3) [쿼리153] 실행
- (4) [0153임시본초용량추출결과_삭제결과 정리] 실행
- (5) [쿼리154] 실행

4) Macro4: 본초-용량 구성 단계

- (1) [031본초용량단위위치차삭제] 실행
- (2) [쿼리31] 실행

- (3) [032위치차_최소값삭제] 실행
- (4) [쿼리32] 실행
- (5) [쿼리33] 실행

최종적으로 Run Macro 명령어를 활용하여 Macro 1,2,3,4를 실행해 방제구성 자동추출을 완성하였다.

3. 결과

본 연구를 진행할 때 PC환경은 Intel Core i7-1065G7 CPU 1.30GHz를 사용하였으며, RAM은 8GB를 사용하였고, Window 10 64비트 운영 체제를 사용하였다. 6,115개 문장에서 19,277개 본초-용량 구성을 추출하였으며 추출하는데 걸린 시간은 1분16초였다.

IV. 결론

본 연구 결과로 문자열의 위치 값을 이용하여 고 의서의 방제 지식을 자동으로 추출하는 방법을 제시하였다. 이 방법을 통해 자동 추출된 방제 구성 데이터는 문헌에 기재된 실제 방제구성과 100% 일치하지 않는다. 하지만 수많은 전적이 기재되어 있는 방제 구성 데이터를 매년 수기로 작성할 수는 없다고 판단된다. 따라서 위 방법을 통해 작성한 방제 구성 데이터를 전문가의 확인을 거쳐 정제하는 과정을 거쳐야 한다. 또한 향후 반복적인 전문가 확인이라는 정제과정을 거친다면 데이터의 오류는 더욱 줄어들 것으로 판단된다.

이와 같은 연구방법은 컴퓨터 프로그램이 읽을 수 있는 데이터로 작성되어 있는 한의서의 경우 인적·물적 자원의 지원을 최소화 하면서도 효율적으로 방제 및 본초의 지식을 추출할 수 있다고 판단된다. 이는 방제 및 본초 용례 연구에서 유용한 연구자료 및 방법으로 사용될 수 있으며, 나아가 처방 구성에서 각 본초가 가지는 비율과 우선순위를 분석하여 각 구성요소와 병증의 관계를 분석하는 등 입체적인 방법으로 방제 구성을 볼 수 있게 한다.

향후 이를 바탕으로 방제-병증, 방제-맥상 병증-

맥상 등의 정보 또한 추출 가능할 것이라 생각하며, 이를 통해 고의서 지식의 유용성이 임상에서 잘 활용될 수 있기를 기대한다.

Reference

1. 박병선, 김은하, 이선아, 이병욱. 『방제학』에 기재된 방제효능과 본초 구성을 기반으로 도출된 효능의 비교 연구. 대한한의학회지 21(1). 2008.
2. 오용택, 김상찬, 이병욱. 본초 구성을 이용한 방제 효능 예측 연구 및 본초의 효능 강도 설정안. 대한한의학회지. 대한한의학회지 21(2). 2008.
3. 김정훈, 이병욱. DB를 활용한 方劑의 類方分析 방법 설계. 대한한의학회지 21(1). 2008.
4. 김현호, 홍효신, 유제혁, 권오민, 차웅석. 객체 지향형 처방데이터베이스의 구축과 처방 검색 프로그램의 설계 및 개발. 한국한의학회지 17(2). 2011.
5. 김기욱, 김태열, 이병욱. 본초 목록을 이용한 방제의 본초 구성 자동 추출 방법. 대한한의학회지 27(3). 2014.
<https://doi.org/10.14369/skmc.2014.27.3.155>
6. 김종현, 배효진, 김창업, 이충열, 신상원. 텍스트마이닝(Text mining)을 활용한 한의학 원전 연구의 가능성 모색. 대한한의학회지. 31(4). 2018.
<https://doi.org/10.14369/jkmc.2018.31.4.027>
7. 彭懷仁 等 主編, 中醫方劑大辭典. 北京. 人民衛生出版社. 2015.
8. 한의학고전DB: <https://mediclassics.kr/>
2020년 10월 16일 검색