

텍스트 내용분석 방법을 적용한 소프트웨어 교육 요구조사 분석: A대학을 중심으로

박금주
단국대학교 교수학습개발센터

The Study on the Software Educational Needs by Applying Text Content Analysis Method: The Case of the A University

Geum-Ju Park

Center for Teaching and Learning, Dankook University

요약 본 연구는 대학생을 대상으로 시행되고 있는 소프트웨어 교육의 강의평가결과에 대해 텍스트 내용분석 방법을 적용하여 수강생의 요구사항을 파악하고 개선방안을 도출하는 데 목적이 있다. 연구방법은 텍스트 내용분석 프로그램을 활용해 단어출현빈도, 핵심단어 선정, 핵심단어의 공출현빈도를 산출하고, 네트워크 분석 프로그램을 활용해 텍스트 중앙성 분석, 네트워크 분석을 실시하였다. 연구결과, 소프트웨어 교육의 좋은 점 네트워크는 ‘교수님’에 대한 언급이 가장 많고 ‘친절’, ‘학생’, ‘설명’, ‘코딩’과 함께 언급되고 있다. 개선점 네트워크는 ‘강의’에 대한 언급이 가장 많고 ‘좋겠다’, ‘학생’, ‘교수님’, ‘과제’, ‘코딩’, ‘어려운’, ‘발표’가 함께 언급되었다. 좋은 점과 개선점에 대한 네트워크 비교 분석에서 공통으로 언급된 핵심 단어 중 조별(활동), 과제, 수업의 난이도, 교수자에 대한 생각에서 차이를 보였다. 이러한 생각 차이는 강의평가 내용을 통해, 개별 조원의 적절한 역할 부족, 어렵고 과다한 과제, 소프트웨어 교육의 난이도와 필요성에 대한 인식, 교수자의 수업방식과 피드백의 부족을 확인할 수 있었다. 따라서, 소프트웨어 교육의 조별(활동)과 과제부여가 어떻게 이루어지고 있는지 살펴보고 강의내용과 교수방법, 실습과 디자인 싱킹을 다루는 비율에 대한 점검이 필요하다.

Abstract The purpose of this study is to understand the college students' needs for software curriculum which based on surveys from educational satisfaction of the software lecture evaluation, as well as to find out the improvement plan by applying the text content analysis method. The research method used the text content analysis program to calculate the frequency of words occurrence, key words selection, co-occurrence frequency of key words, and analyzed the text center and network analysis by using the network analysis program. As a result of this research, the decent points of the software education network are mentioned with 'lecturer' is the most frequently occurrence after then with 'kindness', 'student', 'explanation', 'coding'. The network analysis of the shortage points has been the most mention of 'lecture', 'wish to', 'student', 'lecturer', 'assignment', 'coding', 'difficult', and 'announcement' which are mentioned together. The comprehensive network analysis of both good and shortage points has compared among key words, we can figure out difference among the key words: for example, 'group activity or task', 'assignment', 'difficulty on level of lecture', and 'thinking about lecturer'. Also, from this difference, we can provide that the lack of proper role of individual staff at group activities, difficult and excessive tasks, awareness of the difficulty and necessity of software education, lack of instructor's teaching method and feedback. Therefore, it is necessary to examine not only how the grouping of software education (activities) and giving assignments (or tasks), but also how carried out group activities and tasks and monitored about the contents of lectures, teaching methods, the ratio of practice and design thinking.

Keywords : Network Analysis, Text Analysis, Coding Education, Software Education, College Students

*Corresponding Author : Geum-Ju Park(Dankook Univ.)

Tel: +82-41-550-1734 email: 12171176@dankook.ac.kr

Received December 20, 2018

Revised January 9, 2019

Accepted March 8, 2019

Published March 31, 2019

1. 서론

컴퓨터가 탑재된 기기나 컴퓨터 사용이 보편화된 현재, 컴퓨터 사용법을 익히고 업무에 사용하는 것은 우리의 일상이 되었다[1]. 사회의 발달과 함께 사회 구성원이 갖추어야 할 역량에도 변화가 있는데, 4차 산업사회를 살아갈 미래 세대를 위한 교육은, 우리 사회 운영의 기반인 컴퓨터를 익히고 활용하며 더 나아가 다른 분야에 적용할 수 있는 융합형 인재 양성에 중점을 두어야 한다.

컴퓨터를 익히는 것은 컴퓨터 언어인 코드를 사용해 프로그래밍을 하는 것과 더불어, 프로그래밍을 하기 까지 적절한 사고를 통해 문제를 해결해 나가는 컴퓨팅 사고(computational thinking)가 포함된다. 이렇게 컴퓨팅 사고를 하고 코드를 익혀 프로그래밍 하는 소프트웨어 교육은 초·중등학교와 대학에서 실시하고 있다[2].

대학의 소프트웨어 교육은 문제해결력, 창의적 사고의 배양을 목표로 실시하고 있으며 프로그램으로는 파이선, R, 엔트리, 스크래치, 앱인벤터 등이 활용되고 있다 [3~5]. 교수법으로는 강의식 외에 팀별 과제를 해결하는 팀기반 학습(Team-based learning), 플립러닝(Flipped learning) 등이 적용되고 있다. A대학도 창의적 사고를 향상시킬 수 있는 프로그램으로 구성된 소프트웨어 교육을 실시하고 있다. 3년간 소프트웨어 교육을 시행하면서 활용 프로그램을 교체하고 교재를 개발·활용하며 강의식 외 다양한 교수법 활용, 요구조사 실시 등 효과적이고 체계적인 교육의 실시를 위해 지속적인 노력을 하였다. 이러한 시도에도 불구하고 소프트웨어 교육은 매학기 강의평가의 전체 항목에서 전체 강좌 평점 평균 보다 낮은 분포를 보였다[6]. 강의평가를 통해 교육 수요자인 학습자의 만족도를 파악할 수 있고 서술형 문항으로 기술하게 되어 있는 강의의 좋은 점과 개선점을 통해 수강생의 의견을 수렴할 수 있다. 선택형 문항의 만족도 조사 결과는 수치화된 양적 데이터로 확인이 가능하고 서술형 문항의 텍스트는 텍스트 내용 분석 방법을 적용하여 다양한 형태의 그래프와 수치 데이터로 표현이 가능하다.

텍스트 내용 분석 방법을 활용하여 단어 간 네트워크 분석을 하면 질적 데이터의 양적 분석이 가능하고 수치와 그래프로 시각화하여, 방대한 텍스트 자료에 대한 직관적 파악이 가능하다[7]. 또한 복잡한 관계를 쉽게 설명할 수 있다[8]. 네트워크 분석은 분석대상이 사회, 조직, 개인이었으나 적용범위가 단어와 단어 사이 관계 분석으

로 확장되었다[9].

소프트웨어 교육 강의평가 서술형 문항에 대한 텍스트 내용 분석은 수강생의 요구사항을 파악하고 반영하여 학생 맞춤형 프로그램 실시를 위한 기초자료의 역할을 한다.

각 대학은 최근 2~3년 간 소프트웨어 교육을 실시하면서 효과적인 교육을 위한 다양한 교수법 적용, 교재 개발 및 활용, 활용 프로그램 교체 등 다양한 시도를 통해 시행착오를 겪고 있어 소프트웨어 교육의 조기 정착과 안정적 시행을 위해 수강생의 생각을 살펴 볼 필요가 있다[6].

따라서 본 연구에서는 A대학의 소프트웨어 교육의 좋은 점과 개선점에 대해 텍스트 내용 분석 방법을 적용하고 네트워크 분석을 실시, 소프트웨어 교육에 대한 수강생의 요구를 분석하여 개선안을 제시하고자 한다.

2. 대학의 소프트웨어 교육 운영 현황

최근 2~3년 사이, 각 대학은 컴퓨팅 사고를 할 수 있도록 교육내용과 과정을 마련하고 교양 선택이나 필수로 개설하여 소프트웨어 교육을 실시하고 있다.

선행연구에서 각 대학의 소프트웨어 교육 현황을 살펴보면, B대학은 2016년부터 ‘컴퓨팅사고와 SW코딩’으로 개설하여 전계열의 학생이 수강하도록 하였고 문제해결에 컴퓨팅사고를 적용하고 엔트리를 활용해 구현할 수 있도록 하였다. ‘문제해결과 알고리즘’은 계열별로 특화된 교육으로 인문계 및 예체능계열은 엔트리를 사용하고 자연계는 교육용 플레이봇을 추가로 활용하였다. 교수법으로는 플립러닝, 실습이 활용되었다[3]. C대학은 2016년부터 ‘컴퓨팅적사고’를 개설하여 인문사회계열 학생 대상의 교양선택 과정으로 운영하였고 2017년부터 교양 필수 과정으로 전환하였다. 컴퓨팅 사고에 대한 이론적 내용을 익히고 스크래치, 파이선을 활용해 프로그래밍을 하고 문제해결을 체험할 수 있도록 했다. 블렌디드 러닝(Blended learning) 방식을 도입하여 온라인과 오프라인 강좌를 병행하여 운영하였다[5].

Fig. 1과 같이, A대학도 2016년 1학기부터 ‘창의적 사고와 코딩’으로 개설하여 컴퓨팅 사고를 기반으로 한 소프트웨어 교육을 실시해 오고 있다. 2016년 1, 2학기는 교육내용과 활용 프로그램에 대한 제한없이 교수 재

량에 의한 교육이 실시되었다. 2017년 1학기에는 SAP사의 ERP(Enterprise Resource Planning, 전사적 자원 관리) 프로그램을 도입하고 활용하였다. 교재는 디자인 싱킹(Design thinking)을 기반으로 창의적 사고를 할 수 있는 교육내용으로 구성하였고 전사적 자원 관리 프로그램을 설치하고 활용하는 방법이 포함되었다. 교수법으로는 플립러닝을 도입하여 이러닝 캠퍼스 시스템을 활용해 강의 전 프로그래밍에 필요한 사전지식을 학습하고 본 강의에서 프로그래밍을 하면서 의문점에 대해 질의와 응답으로 진행하였다. 학생들의 수업 활동은 팀별로 주어진 과제를 해결하는 팀기반 학습을 중심으로 이루어졌다.

2018년 1학기부터 신입생 1학년 대상 교양필수과목으로 전환되었고, 프로그램 설치와 활용에 있어서 어려움이 제기되었던 전사적 자원 관리 프로그램 대신 활용이 쉬운 스크래치 프로그램을 활용하였다. 교재는 사회, 인문, 공학, 자연, 예·체능의 5개 계열 특성을 반영하여 개발하였고 2017년에 사용했던 교재에서 디자인 싱킹에 대한 내용이 보강되고 스크래치 프로그램 활용법이 포함되었다. 전체적인 교육과정은 창의적 사고력, 문제해결력 향상을 위한 교육내용으로 구성하였다.

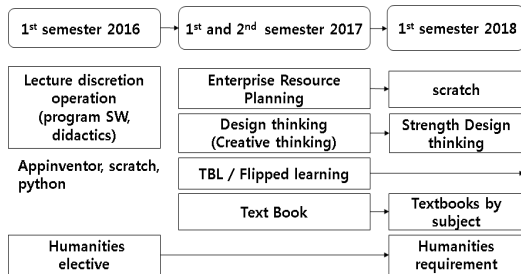


Fig. 1. Status of Software curriculums in A university

지금까지 살펴본 각 대학의 소프트웨어 교육은 공통적으로 2016년 전후, 교양 선택과정으로 시작하여 교양 필수 과정으로 전환하고 교육대상은 일부 계열 학생을 대상으로 제한적으로 실시하다 전 계열 학생으로 확대하였다. 또한 문제해결력과 창의적 사고 향상에 중점을 두고 있으며 오프라인 강의와 온라인 강의를 병행하여 수강생이 오프라인 강의 전 온라인 동영상 강의를 학습하도록 설계되어 있다.

3. 연구방법

3.1 자료수집

A대학의 학사팀에 의뢰하여 소프트웨어 교육 강의평가 결과를 받아 강의평가 문항 중 강의의 좋은 점과 개선점에 대해 텍스트 내용분석 방법을 적용하고 네트워크 분석을 실시하였다. 텍스트 내용분석을 위해 제공받은 자료에서 오타 제거, 한 가지 뜻이지만 여러 단어로 표현된 단어를 통일하고 불용어 제거 등의 전처리 작업은 R 프로그램을 활용하였다.

3.2 분석방법

소프트웨어 교육의 좋은 점과 개선점에 대해 텍스트 내용분석 프로그램인 KrKwic을 활용해 출현단어의 빈도를 계산하고 출현빈도가 높은 단어를 중심으로 핵심단어를 선정하여 KrTitle을 활용해 공출현빈도를 도출하였다[10]. 이렇게 도출된 핵심단어 출현빈도와 공출현빈도를 활용해 UCINET 프로그램에서 네트워크 분석을 실시하였다. 텍스트 중앙성 분석으로 연결중앙성(Degree centrality), 인접중앙성(Closeness centrality), 사이중앙성(Betweenness centrality)을 산출하였다[11, 12]. 연결중앙성은 한 노드가 다른 노드와의 연결 정도, 인접중앙성은 노드와 노드 간 인접 정도, 사이중앙성은 한 노드가 연결망 내의 다른 노드들 사이에 위치하는 정도를 살펴볼 수 있다[9].

마지막으로, 소프트웨어 교육의 좋은 점과 개선점 간 네트워크 비교에서는 각각의 핵심단어 출현빈도를 활용하였다. 자료수집과 분석방법은 Fig. 2와 같다.

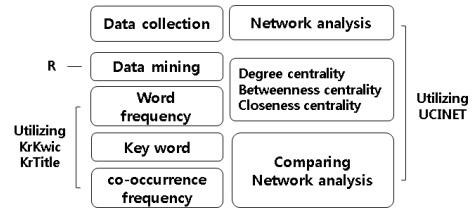


Fig. 2. Data collection and analysis methods

4. 연구 결과

소프트웨어 교육의 좋은 점과 개선점에 대해 핵심단어의 출현빈도를 구하고 핵심단어의 공출현빈도 산출,

텍스트 중앙성 분석, 좋은 점과 개선점 간 네트워크를 비교 · 분석하였다.

4.1 핵심단어의 출현빈도

소프트웨어 교육의 핵심단어는 Fig. 3과 같이, 좋은 점에서는 교수님, 좋다, 강의, 코딩, 친절 등의 순으로, 개선점에서는 강의, 좋겠다, 학생, 과제, 코딩 등의 순으로 나타났다.

소프트웨어 교육의 좋은 점에는 좋다, 친절, 배움, 새로운, 열정, 도움 등의 긍정적 단어가 나타난 반면, 개선점에서는 좋겠다, 어려운, 힘들다, 모르겠다, 개선, 버거움 등의 부정적 단어들이 나타났다. 특히, 강의의 개선을 요구하는 ‘(~하면)좋겠다’의 빈도가 높게 나타났다.

| | Good points | | Shortage points | |
|----|-------------------|-----------|------------------|-----------|
| | Key word | frequency | Key word | frequency |
| 1 | lecturer | 197 | Lecture | 77 |
| 2 | good | 155 | College student | 62 |
| 3 | lecture | 93 | Wish to | 61 |
| 4 | coding | 87 | Assignment | 53 |
| 5 | Kindness | 77 | Coding | 52 |
| 6 | College student | 55 | Difficulty | 42 |
| 7 | Learning | 51 | group activities | 49 |
| 8 | fun | 48 | Lecturer | 40 |
| 9 | Explanation | 41 | Test | 33 |
| 10 | Practice | 39 | Computer | 30 |
| 11 | understanding | 25 | Thought | 23 |
| 12 | Computer | 23 | Hard | 20 |
| 13 | Difficulty | 22 | Contents | 17 |
| 14 | New | 21 | I do not know | 17 |
| 15 | group activities | 19 | Explanation | 17 |
| 16 | Creative thinking | 19 | presentation | 16 |
| 17 | enthusiasm | 17 | Quick progress | 16 |
| 18 | progress | 15 | Understanding | 15 |
| 19 | Assignment | 14 | Progress | 13 |
| 20 | Help | 14 | Scratch | 10 |
| 21 | Scratch | 13 | Program | 9 |
| 22 | Question | 12 | Improving | 9 |
| 23 | Program | 11 | properly | 9 |
| 24 | interest | 8 | Burden | 9 |
| 25 | feedback | 7 | Evaluation | 9 |

Fig. 3. Frequency of occurrence of key words

핵심단어를 활용하여 KrTitle 프로그램에서 공출현빈도를 도출하였다. 소프트웨어 교육의 좋은 점과 개선점에 대해 각각 25개의 핵심어에 대한 공출현빈도를 산출하였다. Fig. 4와 Fig. 5는 전체 공출현빈도 중 각각 10개 핵심단어의 공출현빈도를 제시하였다.

| | Lecturer | Good | Lecture | Coding | Kindness | Student | Learning | Fun | Explanation | Practice |
|-------------|----------|------|---------|--------|----------|---------|----------|-----|-------------|----------|
| Lecturer | 0 | 16 | 11 | 17 | 47 | 38 | 0 | 3 | 21 | 2 |
| Good | 16 | 0 | 9 | 7 | 1 | 6 | 0 | 1 | 2 | 0 |
| Lecture | 11 | 9 | 0 | 3 | 3 | 14 | 0 | 0 | 0 | 0 |
| Coding | 17 | 7 | 3 | 0 | 3 | 8 | 0 | 1 | 5 | 5 |
| Kindness | 47 | 1 | 3 | 3 | 0 | 5 | 0 | 1 | 9 | 1 |
| Student | 38 | 6 | 14 | 8 | 5 | 0 | 0 | 0 | 4 | 1 |
| Learning | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Fun | 3 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Explanation | 21 | 2 | 0 | 5 | 9 | 4 | 0 | 0 | 0 | 3 |
| Practice | 2 | 0 | 0 | 5 | 1 | 1 | 0 | 0 | 3 | 0 |

Fig. 4. The good points of software curriculums: co-occurrence frequency

| | Lecture | student | Wish to | Assignment | Coding | Difficulty | activities | Lecturer | Test | Computer |
|------------|---------|---------|---------|------------|--------|------------|------------|----------|------|----------|
| Lecture | 0 | 37 | 7 | 11 | 10 | 2 | 5 | 23 | 4 | 10 |
| student | 37 | 0 | 12 | 30 | 10 | 2 | 15 | 32 | 1 | 4 |
| Wish to | 7 | 12 | 0 | 11 | 9 | 0 | 7 | 6 | 4 | 3 |
| Assignment | 11 | 30 | 11 | 0 | 4 | 3 | 77 | 23 | 15 | 0 |
| Coding | 10 | 10 | 9 | 4 | 0 | 1 | 1 | 4 | 2 | 9 |
| Difficulty | 2 | 2 | 0 | 3 | 1 | 0 | 0 | 5 | 0 | 0 |
| activities | 5 | 15 | 7 | 77 | 1 | 0 | 0 | 14 | 6 | 0 |
| Lecturer | 23 | 32 | 6 | 23 | 4 | 5 | 14 | 0 | 0 | 0 |
| Test | 4 | 1 | 4 | 15 | 2 | 0 | 6 | 0 | 0 | 0 |
| Computer | 10 | 4 | 3 | 0 | 9 | 0 | 0 | 0 | 0 | 0 |

Fig. 5. The improvement points of software curriculums: co-occurrence frequency

4.2 네트워크 분석

핵심단어 출현빈도와 공출현빈도를 활용해 소프트웨어 교육 강의평가의 좋은 점과 개선점에 대해 네트워크 분석을 실시하였다. Fig. 6과 같이, 좋은 점 네트워크는 ‘교수님’에 대한 언급이 가장 많고, ‘친절’, ‘학생’, ‘설명’, ‘코딩’과 함께 언급되고 있다. Fig. 7과 같이, 개선점 네트워크는 ‘강의’에 대한 언급이 가장 많고, ‘좋겠다’, ‘학생’, ‘교수님’, ‘과제’, ‘코딩’, ‘어려운’, ‘(조별)발표’가 함께 언급됨을 보여준다. 특히, 개선점 네트워크 분석에서 ‘조별(활동)’과 ‘과제’가 함께 언급됨이 두드러진다.

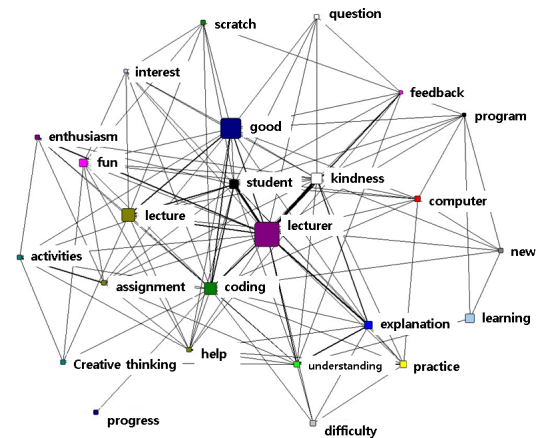


Fig. 6. Network analysis for the good points

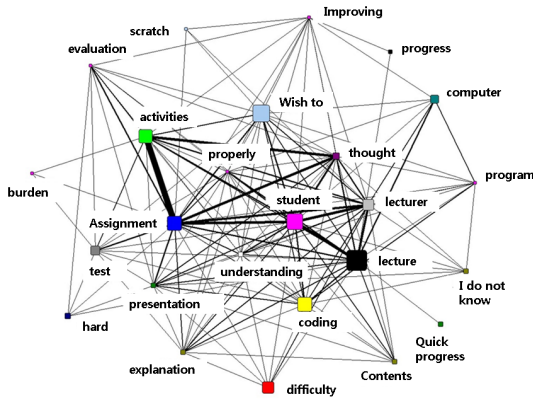


Fig. 7. Network analysis for the shortage points

4.3 텍스트 중앙성 분석

텍스트 중앙성 분석은 공출현빈도를 활용해 소프트웨어 교육의 좋은 점과 개선점으로 나누어 실시하였다. Fig. 8과 같이 좋은 점의 텍스트 중앙성 분석에서, 연결중앙성은 교수님, 학생, 친절, 코딩, 설명의 순이었으며 사이중앙성은 교수님, 이해, 친절, 학생, 좋다는 순이었다. 인접중앙성은 코딩, 교수님, 학생, 친절, 좋다는 순이었다. 개선점의 연결중앙성은 과제, 학생, 조별(활동), 강의, 교수님의 순이었으며 사이중앙성은 과제, 강의, 학생, 교수님, 좋겠다, 조별발표의 순이었다. 인접중앙성은 강의, 학생, 과제, 교수님의 순이었다.

| | Good points | | | Shortage points | | | | |
|----|-------------------|--------|-------------|-----------------|----------------|--------|-------------|-----------|
| | Key word | Degree | Betweenness | Closeness | Key word | Degree | Betweenness | Closeness |
| 1 | Lecturer | 208 | 24.277 | 29 | Lecture | 171 | 15.608 | 27 |
| 2 | Good | 54 | 13.844 | 32 | Student | 209 | 10.054 | 29 |
| 3 | Lecture | 58 | 8.180 | 36 | Wish to | 80 | 7.823 | 32 |
| 4 | Coding | 78 | 18.102 | 28 | Assignment | 257 | 25.604 | 28 |
| 5 | Kindness | 86 | 19.390 | 31 | Coding | 76 | 5.757 | 32 |
| 6 | Student | 110 | 19.371 | 30 | Difficulty | 18 | 0.050 | 40 |
| 7 | Learning | 3 | 0.840 | 51 | Activities | 171 | 6.905 | 34 |
| 8 | Fun | 12 | 4.543 | 39 | Lecturer | 153 | 9.807 | 31 |
| 9 | Explanation | 69 | 1.278 | 39 | Test | 44 | 6.322 | 36 |
| 10 | Practice | 20 | 1.558 | 39 | Computer | 37 | 0.338 | 42 |
| 11 | Understanding | 43 | 19.858 | 35 | Thought | 123 | 4.541 | 32 |
| 12 | Computer | 15 | 2.879 | 39 | Hard | 7 | 0.182 | 43 |
| 13 | Difficulty | 9 | 0.000 | 43 | Contents | 21 | 0.000 | 39 |
| 14 | New | 7 | 3.938 | 42 | I do not know | 11 | 0.125 | 42 |
| 15 | Activities | 16 | 2.080 | 41 | Explanation | 45 | 1.500 | 35 |
| 16 | Creative thinking | 9 | 0.939 | 45 | Presentation | 57 | 7.488 | 32 |
| 17 | Enthusiasm | 24 | 0.435 | 42 | Quick progress | 3 | 0.000 | 50 |
| 18 | Progress | 1 | 0.000 | 51 | Understanding | 61 | 3.983 | 32 |
| 19 | Assignment | 29 | 1.981 | 38 | Progress | 3 | 0.000 | 47 |
| 20 | Help | 25 | 4.432 | 38 | Scratch | 6 | 0.000 | 45 |
| 21 | Scratch | 14 | 1.357 | 41 | Program | 34 | 1.358 | 38 |
| 22 | Question | 16 | 0.000 | 46 | Improving | 22 | 10.875 | 35 |
| 23 | Program | 16 | 6.989 | 42 | Properly | 46 | 4.469 | 32 |
| 24 | Interest | 19 | 0.522 | 41 | Burden | 5 | 0.000 | 49 |
| 25 | feedback | 17 | 0.869 | 42 | Evaluation | 36 | 2.221 | 38 |

Fig. 8. The analysis of text centrality

4.4 네트워크 비교 분석

소프트웨어 교육의 좋은 점과 개선점 네트워크 비교 분석을 살펴보면, Fig. 9와 같이 공통적으로 강의, 수업 진행(과정), 과제, 스크래치, 프로그램, 학생, 컴퓨터, 조

별(활동), 설명, 어렵다, 교수님, 코딩, 이해가 언급되었다. 이 그림에서 숫자는 해당 단어의 출현빈도를 나타내는데, 공통 핵심 단어 중 소프트웨어 교육에 대해 학생들은 ‘조별(활동)’, ‘과제’, ‘수업의 난이도’, ‘교수자’, ‘설명’은 2배 이상의 출현빈도 차이를 보였다.

이러한 생각 차이를 서술형 강의평가 내용에서 살펴보면, 팀기반 학습의 수업진행으로 조별(활동)이 이루어지는데 학생 간 스크래치 프로그램을 다루는 능력 차이가 크거나 적극적으로 참여하지 않는 팀원으로 인해 조별(활동)의 문제점을 언급한 경우가 많았다. 또한, 수업 시간에 실습했던 내용보다 어렵고 과다한 과제가 주어지는 경우를 살펴볼 수 있었다. 수업의 난이도에 있어서는 어렵지만 새로운 것을 배울 수 있어 좋다는 의견과, 어렵고 전공과 무관한 것을 왜 배우는지 모르겠다는 상반된 의견을 확인할 수 있었다. 교수자에 대한 의견에서는 수강생의 수준을 고려하고 배려하는 교수자, 반면에 컴퓨터 실습수업임에도 불구하고 컴퓨터 실습을 진행하지 않는 교수자, 수강생의 질문에 답변이나 피드백을 제공하지 않는 교수자의 상반된 모습을 살펴볼 수 있었다.

좋은 점에서는 질문, 피드백, 흥미, 도움, 새로운, 창의, 열정, 실습, 재미, 좋다, 친절, 배움이, 개선점에서는 버거움, 개선, 제대로, 평가, 모르겠다, 수업내용(과정), 조별발표, 생각, 빠르다, 좋겠다, 시험, 힘들다가 도출되었다.

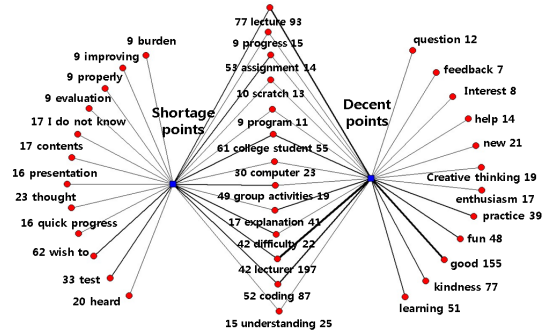


Fig. 9. Comparing network analysis for the good and shortage points

5. 결론 및 논의

A대학의 소프트웨어 교육의 좋은 점과 개선점에 대해 텍스트 내용 분석 방법을 적용한 결과, 소프트웨어 교

육의 좋은 점에서는 ‘교수님’에 대한 언급이 가장 많았고 ‘친절’, ‘학생’과 함께 언급되는 경우가 많았다. 개선점에서는 ‘강의’에 대한 언급이 가장 많았고 ‘조별(활동)’과 ‘과제’가 함께 언급되는 경우가 많았다. 네트워크에서 중요한 위치에 있는 핵심단어를 살펴보면, 좋은 점은 ‘교수님’, ‘학생’, ‘친절’, ‘코딩’, ‘이해’, ‘좋다’가 언급되었고, 개선점에서는 ‘과제’, ‘학생’, ‘조별(활동)’, ‘강의’, ‘교수님’, ‘좋겠다’가 언급되었다.

소프트웨어 교육의 좋은 점과 개선점에 대한 네트워크 비교 분석에서는 조별(활동), 과제, 수업의 난이도, 교수자에 대한 생각 차이를 보였다. 서술형 강의평가 내용을 통해, 팀별 활동에서 팀원이 적절한 역할을 해주지 못하고, 과제는 어렵고 많았으며, 어려운 소프트웨어 교육의 필요성에 의문을 가진 학습자, 교수자의 수업 진행 방법과 적절한 피드백 부족을 확인할 수 있었다.

분석결과를 종합해 보면, 조별 중심의 수업 활동이 이루어지므로 조별 활동에서 과제 수행이 어떻게 이루어지는지 살펴보고, 세부 강의 내용과 교수방법에 대한 점검이 필요하다. 특히, 코딩 실습과 창의적 문제해결력이 조화를 이루어야 하므로 강의내용에 있어서는 디자인 싱킹과 실습을 다루는 비율을 살펴봐야 한다.

References

- [1] J. H. Ku, “A Study on the Software Convergence Education for Non-Majors Computer Science using Creative Robot”, *Journal of the Korea Academia-Industrial cooperation Society*, Vol. 18, No. 2, pp. 631-638, 2017.
DOI: <https://doi.org/10.5762/KAIS.2017.18.2.631>
- [2] Ministry of Education, Ministry of Future Creation Science, Announcement of basic plan to activate software education-Prepare SW education essential and promote school-oriented SW education, 2016.
- [3] H. J. Yoo, “The plan for of Software Education, in Sungkyunkwan University”, *The Korean Association of General Education Conference Catalog*. pp. 157-160, 2015.
- [4] M. S. Lee, “The Challenges of the Software Education for the Students in non-Science and Engineer area”, *The Korean Association of General Education Conference Catalog*. pp. 135-140, 2016.
- [5] Y. S. Kim, “The study on the development of the software courses for the non-related majored students”, *The Korean Association of General Education*, pp. 110-115, 2017.
- [6] G. J. Park, Y. J. Choi, “Exploratory study on the direction of software education for the non-major

undergraduate students”, *Journal of Education & Culture*, vol. 24, no. 4, pp. 273-292, 2018.

- [7] H. R. Min, H. S. Yoon, “A study on Perception differences between Professor & student’s Evaluation of Courses by Openended Reponse: A Network and Content Analysis”, *Journal of Learner-Centered Curriculum and Instruction*, vol. 17, no. 11, pp. 307-320, 2017.
DOI: <http://dx.doi.org/10.22251/jlcci.2017.17.11.307>
- [8] Y. N. Hwang, G. T. Kim, “Comparative study on main research area of national and international universities using network analysis”, *Korean Journal of Comparative Education*, vol. 24. no. 3, pp. 227-244, 2014.
- [9] Y. H. Kim, Y. J. kim, *Social Network Analysis*, Pakyungsa, 2016.
- [10] H. W. Park, Loet Leydesdorff, “Understanding the KrKwic: A computer program for the analysis of Korean text”, *Journal of The Korean Data Analysis Society*, vol. 6, no. 5, pp. 1377-1387, 2004.
- [11] J. W. Kang, J. W. Park, J. H. Kim, “A Network Analysis of Ballistic Technology Keyword”, *Journal of the Korea Academia-Industrial cooperation Society*, vol. 18, no. 4, pp. 311-316, 2017.
DOI: <https://doi.org/10.5762/KAIS.2017.18.4.311>
- [12] K. S. Byun, J. S. Lim, J. W. Park, “Evaluation of Results in Recent Flexible Solar Cell Research Trends via Network Analysis Method”, *Journal of the Korea Academia-Industrial cooperation Society*, vol. 19, no.6, pp. 600-613, 2018.
DOI: <https://doi.org/10.5762/KAIS.2018.19.6.600>

박 금 주(Geum-Ju Park)

[정회원]



<관심분야>

평생교육, 교수법, 학습법

- 2002년 8월 : 단국대학교 교육대학원 전자계산교육(교육학석사)
- 2013년 2월 : 단국대학교 일반대학원 교육학과(교육학박사)
- 2017년 9월 ~ 현재 : 단국대학교 교수학습개발센터 연구교수