소프트웨어 통합 검증을 위한 AI 기반 테스트 절차서 해석 및 로봇팔 제어 명령어 생성 자동화 방안

An AI-Based Approach to Automating Test Procedure Interpretation and Robot Arm Control Command Generation for Software Integration Testing

정주원* · 배찬일** · 양승민** · 이지현***
Juwon Jung* · Chanil Bae** · Seungmin Yang** · Jihyun Lee***

* (주)모아소프트, 광운대학교 대학원 방산 AI 로봇융합학과
** 한국항공우주산업 KFX 임무 SW 팀
*** (주)모아소프트

* wopalw@naver.com

ABSTRACT

This paper proposes an AI-based approach that parses natural-language sentences in software integration test procedures and classifies them into script-based robot-control commands. Procedure texts extracted from Word and Excel files are tokenized and classified into major command categories, achieving a micro-average AUC above 97%. The method reduces manual script entry time and errors, facilitating the deployment of automated testing environments.

Key Words : Robotic Control, Natural Language Understanding, Sequence Classification, Software Integration Testing, BiLSTM, Artificial Intelligence

1. Introduction

현대 항공·방산 분야의 소프트웨어(SW) 통합 검증 과정에서는 방대한 테스트 절차서 문장을 해석해 로봇팔을 제어하는 명령어를 생성해야 한다[1]. 그러나 이 작업은 절차서의 각 문장을 숙련된 인력이 반복적으로 수작업 해석·전환해야 하므로, 작업 시간이 과도하게 소요되고 인적 오류가 빈번하게 발생하는 문제가 있다[2].

본 연구는 Word·Excel 형식의 절차서 텍스트를 추출·정제·토큰화한 뒤, BiLSTM 기반의 분류기를 이용해 네 가지 명령어 유형으로 자동 분류하고, 결과를 로봇팔 제어 script 형태로 구조화하는 AI 기반 방법론을 제안한다. 이를 통해 수작업 Script 입력에 소요되는 시간과 오류를 획기적으로 줄이고, 자동화 테스트 환경 구축의 효율성을 확보할 수 있다.

본 논문의 주요 기여는 다음과 같다.

1. 실제 현장 절차서 기반의 실용적 데이터셋 구축
2. 일관적이고 최적화된 전처리·임베딩 파이프라인
3. BiLSTM 분류기를 활용해 네 가지 명령어 분류 및 97% 이상의 Micro-average AUC 달성
4. 분류 결과를 로봇팔 제어용 Script 명령어로 자동 변환하는 메커니즘 개발

2. Related Works

기존 테스트 자동화는 정규 표현식 및 도메인 규칙 기반으로 명령어를 추출했으나[3], 새로운 표현에 취약해 일반화가 어렵다는 한계가 있다.

최근 Transformer 및 RNN 계열의 분류기를 활용해 테스트 절차서 문장을 자동 분류하는 연구가 증가하고 있으며, 특히 BERT·LSTM 계열이 소량 데이터에서도 안정적인 성능을 보이는 것으로 보고되고 있다[4,5]. 그러나 도메인 특화된 로봇 제어 명령어를 자동 생성한 사례는 아직까지 드문 실정이다[6]. 이에 본 연구는 실제 절차서 데이터를 기반으로 BiLSTM 모델을 활용한 문장 자동 분류 및 제어 Script 생성 방안을 제안한다.

3. Proposed Method

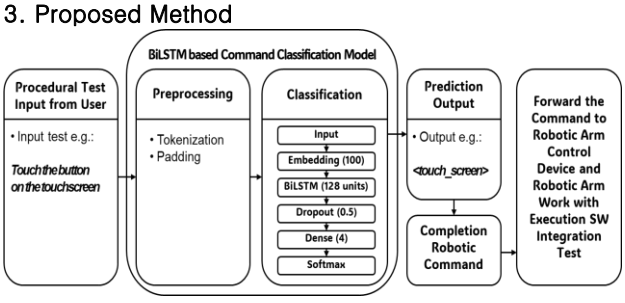

Fig. 1. System Workflow and BiLSTM-based Model Construction

Fig. 1.에 나타낸 전체 프로세스는, 먼저 사용자가 입력한 자연어 테스트 절차 문장을 토큰화·패딩하여 고정 길이 시퀀스로 변환한 뒤, Embedding → BiLSTM (→ Dropout 및 Dense → Softmax) 분류 과정을 거쳐 <verify_patch_value>, <verify_image>, <touch_screen>, <custom_command> 등의 제어 태그로 예측한다. 이후 분류된 태그를 기반으로 제어 Script를 자동 생성하여 로봇 제어 장치에 전달하면, SW 통합 검증 환경에서 로

봇팔이 해당 제어 Script 명령어를 실행한다.

3.1 Data Collection and Preprocessing

Word·Excel 형식에서 수집된 실제 테스트 절차서 문장 총 395건을 전처리하여, 테스트 절차를 'Input'으로, 제어 Script 명령어 태그를 'Target'으로 라벨링하였다. 'Target'의 Opening 태그로써, <verify_patch_value>, <verify_image>, <touch_screen>, <custom_command> 네 가지 주요 명령어 클래스를 정의하였다.

3.2 Sequence Encoding and Input Representation

문장을 Tokenizer로 단어 시퀀스로 변환하고, 최장 문장 길이에 맞춰 패딩을 거쳐 입력 벡터를 구성하였다.

3.3 BiLSTM-Based Command Classification Model

Embedding 레이어(100차원)를 통해 시퀀스를 벡터로 변환하고, BiLSTM(128 units) 계층을 통해 문맥 정보를 추출하였다. 과적합을 방지하기 위해 Dropout(0.5)을 적용하였으며, Softmax를 통해 각 명령어 유형에 대한 확률을 예측하였다. 모델 학습은 Adam 옵티마이저와 categorical_crossentropy 손실함수를 사용하며, EarlyStopping을 적용해 학습 안정성을 확보하였다.

4. Results and Discussion

4.1 Command Tagging Examples

예측 결과는 Opening/Closing 태그 형태로 출력된다. 예를 들어, "터치스크린의 버튼을 터치한다."는 <touch_screen>로 분류되며, 대응하는 Closing 태그는 </touch_screen>로 자동 생성된다.

4.2 Overall Performance Evaluation

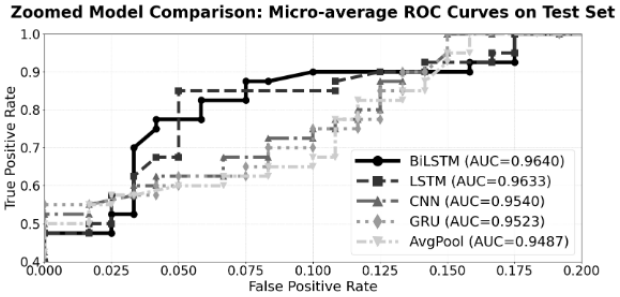

Fig. 2. AI Model Performance Comparison

전체 395개의 샘플을 학습(90%)·테스트(10%)로 분리하고, 학습셋 내에서 10%를 stratified 검증셋으로 추가 분할하였다. 테스트 결과는 다음과 같다.

- **BiLSTM Macro-average AUC:** 96.40% (주로 96%~98% 범위)
- **모델별 Micro-average AUC 분석:** 테스트 세트에 대한 Micro-averag ROC 곡선을 FPR 0.0~0.2, TPR 0.4~1.0 영역으로 확대(Zoom)하여 Fig. 2.와 같이 비교하였다. AUC는 BiLSTM(0.9640), LSTM(0.9633), CNN(0.9540), GRU(0.9523), AvgPool(0.9487)로 LSTM 계열의 성능이 가장 높았으며, 그 중 BiLSTM이 가장 성능이 높았다. 낮은 FPR 구간에서도 높은 TPR을 유지함으로써, 실제 적용 시 오탐(False Positive)을 최소화하면서도 정확한 분류 성능을 확보할 수 있었다.
- **정확도(Accuracy):** 83.80%
- **Macro-average F1-Score:** 83.12%
- **Macro-average 재현율(Recall):** 89.33% (클래스별 편차 존재, 특히 <custom_command>에서 59.87%로 최저)
- **혼동 행렬 분석:** <custom_command>의 약 40%(61/152개)가 <verify_image>로 오분류됨

4.3 Analysis of Errors and Model Limitations

BiLSTM 기반 모델의 전체적인 정확도는 우수하였으나, 실제 데이터가 제한적이어서 클래스 불균형에 따른 편향이 관찰되었다. 특히 <touch_screen> 유형은 전체 데이터가 2건에 불과했고, <custom_command> 중 약 40%가 <verify_image>로 오분류되어 정확도가 상대적으로 낮아졌다. 따라서 향후 소수 클래스에 대한 데이터 증강 및 가중치 기반 손실함수 적용 등이 필요하다.

5. Conclusion

본 연구는 SW 통합 검증 절차서 문장을 BiLSTM 기반 AI 모델로 자동 해석하고, 로봇팔 제어 명령어 유형으로 분류한 뒤, 제어 Script를 자동 생성하는 방법을 제안하였다. 연구 결과, Macro average AUC 약 96.40%, Accuracy 83.80%를 기록하였으며, 다섯 가지 모델(BiLSTM, LSTM, CNN, GRU, AvgPool) 중 BiLSTM이 가장 우수한 성능을 보였다. 제안한 본 방안은 Script 수작업 입력 시간과 오류를 줄여, 현장 작업의 효율성과 안전성을 향상시킬 수 있다. 향후, 데이터셋 확장, 클래스 불균형 완화, Transformer 기반 모델 적용 등을 통해 성능 개선이 기대된다.

References

[1] D. Banerjee, and K. Yu, "Integrated Test Automation for Evaluating a Motion-Based Image Capture System Using a Robotic Arm," IEEE Access, pp. 1-1, 2018.

[2] V. Garousi, S. Tasli, O. Sertel, M. Tokgoz, K. Herkiloglu, H. F. E. Arkin, and O. Bilir, "Automated Testing of Simulation Software in the Aviation Industry: An Experience Report. IEEE Software," vol. 36, no. 4, pp. 63-75, 2019.

[3] Z. Zhan, "Use Regular Expressions in Test Automation," The Agile Way, 2022.

[4] A. Ezen-Can, "A Comparison of LSTM and BERT for Small Corpus," arXiv, 2020.

[5] W. Huang, Y. Sun, X. Zhao, J. Sharp, W. Ruan, J. Meng, and X. Huang, "Coverage guided testing for recurrent neural networks," arXiv, 2019.

[6] H. Luo, J. Wu, J. Liu, and M. F. Antwi-Afari, "Large language model-based code generation for the control of construction assembly robots: A hierarchical generation approach," Developments in the Built Environment, vol. 19, 2024.