

## 생성형 AI 기반 프로그래밍 지원도구의 통역 실무 응용 및 통역사 역할 확대 가능성 고찰: 용어 추출 및 기계통역 웹애플리케이션 개발을 중심으로

허지운\*

*Jiun Huh (2025). Exploring the potential of using AI programming assistants for supporting interpreters' professional needs and expanding their roles: Developing application prototypes for terminology extraction and machine interpreting. This study focuses on developing interpreting application prototypes using an AI programming assistant, exploring how such technologies can support interpreters' professional needs and potentially expand their roles. It adopts the vibe coding paradigm, which involves generating code through natural language prompts. Using Cursor, an AI code editor, the study developed three prototypes: (1) a terminology extraction web application, and (2) two machine interpreting web applications. The findings demonstrate how AI programming assistants can be leveraged to create custom solutions that support interpreting workflows while also helping interpreters build technological literacy. The study also discusses the strengths and limitations of current machine interpreting technologies and the potential professional opportunities they present for interpreters. Finally, it highlights the fundamental nature of communication — encompassing both linguistic and non-linguistic features — and emphasizes the importance of human-technology coexistence through collaborative partnerships. (Ewha Womans University, Korea)*

---

\* 이화여자대학교, 부교수, 제1저자

**Keywords: AI programming assistant, vibe coding, interpreter, automatic speech translation, machine interpreting**

**주제어: AI 프로그래밍 지원도구, 바이브코딩, 통역사, 자동음성번역, 기계통역**

## 1. 서론

생성형 AI가 급속도로 우리 일상에 뿌리를 내리고 있다. 2022년 OpenAI의 GPT-3.5 발표와 함께 인공지능은 가파른 속도로 사회 전반에 확산하고 있다. GPT-3.5의 자연스럽고 유창한 대화 능력은 전 세계에 충격을 주었고(이종현 2023; Wiggers et al. 2025), 이후 발표된 GPT-4o는 정밀한 논리적 사고까지 수행하며 미국의 각종 대학 입시 및 변호사 자격시험(SAT, GRE, LSAT 등)에서 상위 10~20%에 해당하는 고득점을 획득하면서 전문직 영역에까지 위기감이 확산하였다(Edwards 2023). 자연스럽고 유창한 대화를 구사하고 번역 능력까지 겸비한 GPT-3.5가 공개되었을 당시 언론과 소셜미디어에서는 인공지능에 의해 대체될 직업을 주제로 한 콘텐츠가 쏟아졌으며, 통번역사도 그 대상에 포함되어 있었다. 그러나 GPT-4o 출시 이후로는 거의 모든 화이트칼라 전문직이 위기감을 느끼게 되었고, 인간 노동 전반의 대체 가능성에 대한 우려가 한층 확산되었다(Briggs and Kodnani 2023; Kelly 2025).

그러나 과연 언론과 소셜 미디어의 주장처럼 위기가 현실화 되었을까? 생성형 AI가 인간의 노동을 대체할 것이라는 주장이 적용되는 범위는 어디까지일까? 이것이 통번역 현장에 미치는 함의는 무엇일까? 과거의 궤적만으로 미래를 예단할 수는 없겠지만, 역사는 비슷한 흐름을 반복한다는 점에서 통번역 관련 기술의 등장과 이후의 궤적을 검토해 볼 필요가 있다. 통번역 역사에서 두드러진 기술 혁신으로는 인간 통역사의 동시통역 프로세스를 지원하는 동시통역 모듈을 들 수 있다. IBM에서 1927년 최초로 개발한 이 기술 덕분에 순차통역으로만 진행되던 회의에 동시통역의 가능성이 열렸다(IBM n.d.). 이후 IBM은 이 기술을 발전시켜 무선 기술 기반의 동시통역 시스템인 IBM Wireless Translation System을 개발하였고, 이를 UN, 올림픽 위원회, 적십자, 뉘른베르크 국제 군사재판에 공급하면서 동시통역이 널리 확산하는 계기가 되었다(IBM n.d.). 번역에 있어 획기적인 기술은 단연 컴퓨터보조번역 도구인 CAT(Computer Assisted Translation)이다. 원문과 번

www.kci.go.kr

역 코퍼스가 쌍으로 정렬되어 번역 메모리에 기록됨에 따라, 반복되는 원문에 대한 기존 번역 결과물을 재활용할 수 있게 되어 번역 시간 단축 및 번역 물량 확대에 기여하게 되었다(이준호와 김순미 2022; Pangnani 2023).

한편, 기계통역(Machine Interpreting) 또는 자동음성번역(Automated Speech Translation, AST)에 대한 관심과 개발 노력은 2000년으로 거슬러 올라갈 수 있다(허지운 2016: 174). 인간 통역사를 건너뛰고 서로 다른 언어를 구사하는 발화자 간에 각자의 모국어로 대화를 할 수 있도록 한다는 취지로 개발이 시작되었지만, 그동안은 공허한 구호에 그치고 만 것이 현실이며, 국제회의 및 통역 시장이 성장을 거듭하는 가운데 기계통역은 존재감을 보이지 못하였다(허지운 2016). 이처럼 그간에 등장한 획기적이면서 유효한 통번역 기술은 인간 통번역사의 업무 효율을 제고하고 신규 시장 확대에 기여하는 방향으로 발전해 왔다. 이는 궁극적 서비스 공급자인 인간 통역사의 편의성과 효율성을 증대하는 유용한 ‘보조도구’로 기능해 왔기 때문이다. 생성형 AI는 인간이 그동안 차지해 온 서비스 공급 주체의 역할을 대체할 수 있다는 위기감을 준다는 점에서 과거의 통번역 보조 기술과 차별된다. 그러나 이러한 위기감은 인간과 기술을 대립 구도로 바라보는 인식에 기인한다. 지금까지는 이 구도가 이어져 왔다. 인간 통번역사는 언어, 문화, 커뮤니케이션 전문가이지 기술 전문가가 아니고 그럴 필요도 없었기 때문이다. 그러나 최근 생성형 AI의 발달은 소프트웨어 프로그래밍의 초기 진입장벽을 낮추는 변화를 수반하였다. ChatGPT에 자연어로 프롬프트를 입력하여 대화형으로 번역을 주문하듯, 생성형 AI에 대화형 프롬프트를 입력하여 간단한 기초 코딩도 수행할 수 있게 된 것이다. 카르파티(Karpathy 2025)가 ‘바이브 코딩(vibe coding)’이라 칭한 생성형 AI 기반 프로그래밍 패러다임은 통번역사의 기술 이해에 일조할 가능성을 제시해 주고 있다. 이에 본고에서는 실제 바이브코딩을 실시하여 생성형 AI 기반 프로그래밍 보조도구가 통역사의 실무에 어떻게 유용하게 응용될 수 있고 통역사 역할 확대에 있어 어떠한 가능성을 제시하는지 고찰해 보고자 한다.

## 2. 이론적 배경

바이브코딩(vibe coding)은 테슬라(Tesla)의 전 AI 부문 이사이자 OpenAI의 창립멤버인 안드레이 카르파티(Andrej Karpathy)가 감이나 직관적 흐름에 따라 코딩

www.kci.go.kr

을 하는 새로운 프로그래밍 패러다임을 명명한 것으로, 거대언어모델(Large Language Model, LLM)의 지수함수적 발전에 힘입어 자연어 프롬프트 입력만으로 코딩하는 새로운 방식을 지칭한다(Karpathy 2025). 화이트(White 2024)는 생성형 AI의 기반인 트랜스포머 모델(Transformer model)이 본래 언어 번역 용도로 구축되었지만, 입력된 프롬프트를 분류, 해석, 실행하는 과정도 결국 입력자의 ‘의도’를 ‘번역’하는 과정이라고 보면서, 생성형 AI가 이러한 ‘번역’ 과정을 통해 인간의 의도와 컴퓨터의 작동 간 간극을 좁히는 혁신을 불러올 것이라고 주장하였다. 다시 말하면, 사용자가 자연어 프롬프트를 입력하면 생성형 AI가 입력한 사람의 의도를 자연어 문장 속에서 파악하여 이를 소스코드로 번역하고, AI 에이전트(Agentic AI)가 기존 소프트웨어 툴의 다양한 API와 연동하여 입력자가 의도한 목표를 달성하는 것이 가능하다는 것이다(White 2024). 화이트는 이것이 정적인 코딩이 아닌 동적이며 살아 있는 코딩이라고 주장하며, 이 과정에서 지적재산권은 코드 그 자체가 아닌 입력자의 목표와 의도가 담겨 있는 프롬프트 대화에 부여되어야 한다고 보았다. 대표적 바이브코딩 도구, 또는 AI 기반 프로그래밍 보조 도구로는 Cursor, GitHub Copilot, Replit Ghostwriter를 들 수 있다.

바이브코딩은 코드를 우선 생성한 후 반복적 프롬프트 입력을 통해 세밀하게 정리하는 방식이기 때문에, 개발자는 본격적인 실행에 앞서 초기 컨셉과 아이디어를 프로토타입으로 손쉽게 제작하여 시험해 볼 수 있고, 바이브코딩 툴의 자동완성 기능을 이용하면 반복적으로 재사용되는 보일러플레이트 코드(boilerplate code) 작업을 줄일 수 있어, 개발자의 편의성과 생산성을 증대시킬 수 있다(Barke et al. 2023; Sajja et al. 2024; Harkar 2025). 또한 개발자에게는 자신이 잘 모르던 프로그래밍 언어나 기술을 빠르게 습득할 기회가 된다(Smith 2025a). 여기에서 한발 더 나아가, 이렇게 낮아진 진입장벽 덕에 코딩 문외한조차 바이브코딩으로 앱을 만들 수 있다(Harkar 2025). 사용자가 프롬프트에 원하는 목표를 자연어로 입력하면 해당 생성형 AI 프로그래밍 보조도구가 코드를 생성하고, 이용자는 코드가 산출한 앱을 확인하며 대화 프롬프트를 통해 부족한 부분을 다듬으면 될 정도로 이용법이 간편하기 때문이다(Fakhoury et al. 2024). 가장 매력적인 프로그래밍 언어는 영어라고 카르파티가 표현할 정도로 손쉽게 대화형 코딩이 가능하다는 것인데(Karpathy 2023), 바이브코딩 툴인 Cursor나 GitHub Copilot의 프롬프트는 영어뿐 아니라 한국어를 비롯한 다양한 언어도 지원한다.

실제로 개발자들도 현장에서 바이브코딩이 가능한 생성형 AI 프로그래밍 보조

도구를 사용하고 있다. GitHub의 미국 개발자 대상 설문조사에서 92%가 이미 업무 내외적으로 AI 코딩 툴을 사용하고 있고 70%는 코드 품질과 성능 향상의 장점이 있다고 응답하였다(Shani 2023). 전문 개발자가 아닌 일반인도 바이브코딩을 시도하고 있다. ChatGPT, DallE, Midjourney를 이용하여 3일 만에 앵그리버드와 비슷한 게임을 개발한 사례(Lopez 2023), ChatGPT와 Cursor를 이용하여 PDF 문서를 마크다운 형식으로 일괄변환하는 자동화 스크립트를 10분 만에 개발한 사례(McNulty 2025), Claude 3.5 Sonnet을 이용하여 간단한 주사위 게임을 개발한 사례(Smith 2025b)를 통해 코드를 알지 못하여도 생성형 AI와의 지속적 프롬프트 대화를 통해 간단한 앱을 손쉽게 개발할 수 있음을 알 수 있다.

그러나 손쉬운 만큼 분명한 한계도 있다. 스택오버플로우(Stackoverflow 2023)의 9만 명의 개발자 대상 2023년 설문조사에서 77%의 응답자가 생성형 AI 프로그래밍 보조도구에 대하여 호감을 보였지만 42%는 정확도 면에서 신뢰하지 않는다고 응답하여 개발자들이 생성형 AI 프로그래밍 보조도구의 한계를 체감하고 있음을 알 수 있다. 다빌라 외(Davila et al. 2024)가 브라질의 개발자들을 대상으로 면접과 설문조사를 통해 생성형 AI 프로그래밍 보조도구에 대한 인식을 조사한 결과 개발자들은 ChatGPT와 GitHub Copilot을 이용하여 개발하였으며 해당 도구가 생산성 향상에는 도움이 되지만 맥락에서 벗어난 코드를 제시하거나 기존 레거시 도구와의 호환성이 떨어지는 등의 한계가 있다고 지적하였다(Davila et al. 2024). 바이틸링엄 외(Vaithilingam et al. 2022)는 GitHub Copilot을 이용한 코딩 과제 비교 실험을 시행하였는데, Github Copilot이 비교적 정확한 코드를 생성하고 코딩의 유용한 출발점을 제시하며 온라인 검색 시간을 단축해 주는 장점은 있었지만, 그것이 실제 코딩 시간 단축으로 이어지지 못하는 못하였다고 보고하였다. 이는 AI가 자동으로 생성한 코드의 의도를 참가자들이 이해하고 수정하거나 숨은 버그를 찾아내 디버깅하는 것이 어려웠기 때문이다(Vaithilingam et al. 2022). 와이즈 외(Weisz et al. 2025)는 IBM 사내에 배포된 LLM 기반 Watsonx Code Assistant를 사용한 684명의 개발자를 대상으로 설문조사와 사용성 테스트를 실시한 결과, 대다수의 참가자가 생산성 향상을 체감하였으나 속도와 품질은 기대에 못 미쳤고 AI가 생성한 코드의 저작권 및 책임 소재가 불명확하다고 응답한 것으로 나타났다.

전술한 한계에도 불구하고 바이브코딩이 코딩의 진입장벽을 낮추고 코딩 아이디어의 출발점을 제공하며 신속한 프로토타이핑을 가능케 하는 장점을 고려하면 코딩 초급자 교육에 긍정적 시사점을 제공한다고 볼 수 있다. 판 외(Fan et al.

2025)는 대학교 학부 과정 코딩 수업에서 GPT-3.5 Turbo 및 Claude Opus를 사용한 학생이 그렇지 않은 학생에 비하여 프로그래밍 불안감 감소, 코딩 동기부여 상승, 코딩 과제 성과 향상 효과를 보여 초보자에게 유용한 학습 도구로 기능할 수 있음을 시사하였다. 썬 외(Sun et al. 2024)에서는 ChatGPT를 이용한 코딩 학습 시 학습자들이 오류 로그 확인과 디버깅을 더 활발히 하고 웹사이트에 콘솔 메시지를 복사하여 피드백을 구하는 빈도가 더 높았으며, 평균 성적도 유의미한 차이는 아니나 향상된 것으로 나타났다. 또한 학습자들이 생성형 AI를 활용한 코딩의 장단점을 명확히 인식하는 계기가 되었고 문제해결 능력도 개선된 것으로 나타났다(Sun et al. 2024). 선행연구를 통해 확인하였듯 생성형 AI의 확산과 함께 최근 새롭게 부상한 프로그래밍 패러다임인 바이브코딩은 신속한 프로토타이핑, 낮은 코딩 진입장벽 및 코딩 학습곡선 향상의 장점을 지니고 있다. 이러한 특성은 기술 활용을 요구받는 통역사에게 실무나 교육 측면에서 유익한 기회로 작용할 가능성이 있다. 이에 본고에서는 생성형 AI 프로그래밍 보조도구를 이용하여 실제 통역 관련 앱을 개발해 보면서 바이브코딩과 통역 관련 기술의 장단점을 검토하고, 통역사 실무 응용 및 통역사 역할 확대 가능성을 고찰해 보고자 한다.

### 3. 웹 애플리케이션 프로토타입 개발

#### 3.1. 연구 방법

본 연구에서는 생성형 AI 기반 프로그래밍 도구를 이용하여 바이브코딩 패러다임에 따라 자연어 프롬프트 기반 프로그래밍을 수행하여 통역 관련 애플리케이션 프로토타입을 개발하고 사용성 테스트를 실시함으로써, 현재 통역 관련 기술의 장단점을 파악하고 통역사 역할 확대에 관한 함의를 도출하고자 한다. 이에 본 연구에서는 통역 관련 도구를 프로토타입 형태로 개발하고자 하며, 프로토타입 개발을 위해 생성형 AI 기반 프로그래밍 보조도구 중 Cursor 0.49.6 버전을 선택하였다. Cursor는 AI 코드 편집기(AI code editor)를 표방하며 코드 자동완성 및 프롬프트를 통한 AI 기반 자동코딩을 제공한다.<sup>1)</sup> 연간 반복매출(ARR)이 2023년 1백만 달

---

1) <https://www.cursor.com/features>

리에서 2024년 1억 달러로 12개월 만에 급증하며 폭발적으로 성장한 Cursor는 주요 구독 고객층이 개인 개발자들이라는 점에서(Asplund 2025; Tap Twice Digital 2025), 본 연구의 개발 목적 및 환경에 적합하다고 판단하여 선택하였다. Cursor를 이용한 프로토타입 개발을 위하여 월 20달러의 유료 프로(Pro) 요금제를 구독하였다. 그림 1은 본 연구에서 사용한 Cursor 애플리케이션의 프롬프트 기반 코딩 방식을 보여주는 예시 화면이다.

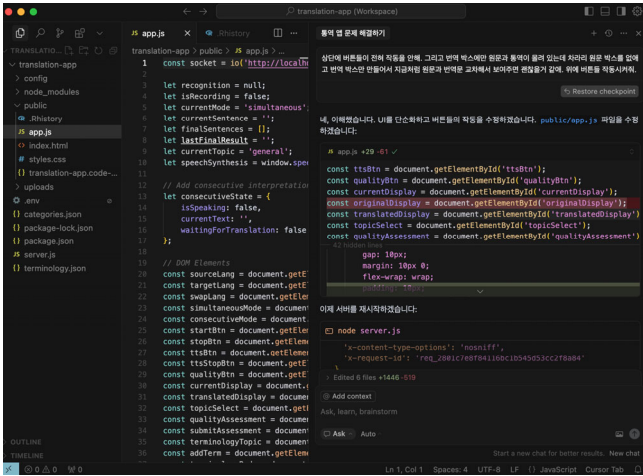


그림 1. Cursor의 프롬프트 기반 코딩 화면

이와 더불어 코드에 대한 설명 참고 및 프로그램 개발을 위한 로직 아이디어 탐색을 위하여 GPT-4o 및 GPT-o3 유료 버전을 사용하였다. 프로그래밍은 ChatGPT 등 범용 생성형 AI 사용 방식인 자연어 프롬프트 입력 방식을 사용하였고, 직접 코드 편집 작업은 전혀 수행하지 않아 순수한 바이브코딩 패러다임을 적용하였다고 할 수 있다. 본 연구에서는 통번역 관련 도구인 용어 추출 프로그램과 기계통역 프로그램 프로토타입을 로컬 호스트 기반의 데스크탑 웹 애플리케이션 형태로 구축하였다. 이는 통역 보조 도구의 실용성과 통역 자동화 도구의 화제성을 고려한 것으로, 용어추출 프로그램은 통역 실무 준비의 핵심인 용어 정리 및 학습 단계에 기술을 접목하면 통역 보조에 도움이 될 수 있다는 점에서(Fantinuoli 2017), 그리고 기계통역 프로그램은 최근 통번역 관련 프로그램 중 주목을 받고 있으나 그동안 개별 통번역사가 직접 구현해 보기 어려운 기술이었다는 점에서 선

정하였다.

### 3.2. 용어 추출 프로그램

본 연구에서는 통역사의 언어 역량 함양을 위한 용어 추출 웹 애플리케이션을 개발하였다. 통역사는 통역 역량을 함양하거나 통역 실무 과제를 앞두고 사전 준비에 매진한다. 통역 과제에 앞선 사전 준비는 언어 및 언어외적 지식의 격차를 좁히고 통역 수행 시 인지 부하를 줄이기 위해 중요하다(Fantinuoli 2017). 이에 통역사는 준비 과정에서 다양한 기본 텍스트 및 도메인 정보가 담긴 텍스트를 읽으며 어휘력, 문장력 및 텍스트 이해력을 함양한다. 특히 전문분야의 경우 일반 명사, 명사나 명사구 중심의 전문용어뿐 아니라 술어적 특성을 지닌 동사와 형용사, 그리고 명사와 고빈도로 호응하는 동사를 숙지할 필요가 있다(이미경 2018; 최문선 2018, 2019). 판티누올리(Fantinuoli 2017)는 통역사가 특정 분야 통역 경험이 풍부 한지와 무관하게 전문용어뿐 아니라 해당 도메인에서 자주 쓰이는 일반 용어들도 익혀야 한다고 지적하며, 전문분야 통역을 위해 익혀야 할 용어를 고빈도 전문용어, 저빈도 전문용어, 도메인 특화 일반 용어로 분류하였다. 그리고 통역사의 용어 습득 및 용어집 관리의 중요성을 강조하며, 통역사가 이 과정을 자율적이고 용이하게 진행하는 방법으로 코퍼스 기반 통역 준비(Corpus Driven Interpreter Preparation, CDIP)와 데이터 주도 학습(Data Driven Learning, DDL)을 제안하였으며, 이를 기반으로 용어와 연어 추출 기능, 해당 표현이 어떤 문장과 맥락 속에서 사용되었는지를 보여주는 콘코던서(Concordancer), 그리고 해당 용어의 번역 표출 기능을 구현한 프로그램을 제안하였다(Fantinuoli 2017).

상기의 내용을 참고하여 본고에서는 학습용 및 통역 준비용으로 활용할 수 있는 용어추출 프로그램 프로토타입을 개발하였다. 물론 이미 인터프리터스 헬프(Interpreter's Help),<sup>2)</sup> 인터프리트뱅크(InterpretBank),<sup>3)</sup> 스케치엔진(Sketch Engine)<sup>4)</sup> 등의 상용 용어 추출 및 관리 서비스들이 있는 것도 사실이다. 그러나 해당 도구들은 개인 맞춤형 도구가 아니고, 서비스 가입을 해야 하며, 한시적 무료 시범 사용 기간이 지나면 유료 서비스를 이용해야 한다. 개인 맞춤형 용어 추출 프로

---

2) <https://interpretershelp.com>

3) <https://www.interpretbank.com/index.html>

4) <https://www.sketchengine.eu>

그램 개발 시 사용자 니즈에 맞춤형된 기능을 설계할 수 있고, 개인 용어집을 상용 서비스 클라우드에 업로드할 필요가 없어 개인 데이터 통제나 보안에 대한 우려가 덜하다. 더 나아가 프롬프트 기반의 개인 맞춤 프로그램을 개발하는 과정에서 AI 작동 방식에 대한 이해를 심화시킬 수 있어 통역사의 체험 기반 AI 리터러시 향상에 도움이 될 것으로 사료된다.

### 3.2.1. 최초 설계 및 구현

연구자가 당초 설계한 것은 사용자가 도구에 입력한 영문 텍스트에서 주요 전문 용어와 표현들을 추출하고 그에 상응하는 한국어 대응어를 번역하여 병기하는 용어집이었다. 이 요구사항에 대하여 Cursor는 TypeScript와 React를 이용한 기본적인 용어추출 웹 애플리케이션 구조를 생성하였다. 이후 품사별 용어 추출과 불필요한 기능어(전치사, 관사/정관사, 숫자 단독 추출) 제외를 요청하는 프롬프트를 입력하여 관련 작업도 진행하였다. 웹 애플리케이션은 로컬 호스트(<http://localhost:3003/>)에서 실행하였다. 웹 브라우저는 MacOS 15.4.1에서 구동되는 Chrome 135.0.7049.115 버전을 이용하였다. 아래 표 1은 연구자의 프롬프트와 Cursor의 코딩 내용이다.

표 1. 프롬프트 기반 1차 프롬프트 및 코딩 내용

최초 프롬프트	<ul style="list-style-type: none"> <li>• 용어추출 앱 생성 요청</li> </ul>
코딩 진행 내용	<ul style="list-style-type: none"> <li>• package.json 파일 생성하여 프로젝트의 의존성을 설정</li> <li>• tsconfig.json 파일 생성하여 TypeScript 설정 추가</li> <li>• scr 디렉토리 및 기본 컴포넌트 생성</li> <li>• 용어 타입 정의 파일 생성</li> <li>• 메인 앱 컴포넌트 생성</li> <li>• index.tsx 파일 생성하여 앱 렌더링</li> <li>• public 디렉토리 생성 및 index.html 생성</li> <li>• 필요 패키지 설치 및 개발 서버 실행</li> </ul>
구현된 기본구조	<ul style="list-style-type: none"> <li>• 용어 목록 그리드 형태 표시</li> <li>• 각 용어 카드에 용어, 정의, 카테고리 표시</li> <li>• 반응형 디자인</li> <li>• 모던한 UI/UX</li> </ul>
추가 프롬프트	<ul style="list-style-type: none"> <li>• 품사별 주요 용어 추출 (자연어 처리, 문맥 분석)</li> <li>• 기능어(관사, 정관사, 전치사, 의문사, 접속사, 일반적 어휘) 단독,</li> </ul>

	숫자, 윌, 요일, 문장부호 등 제외 <ul style="list-style-type: none"> <li>• 전문용어 추출 강화(명사형, 복합어 위주)</li> <li>• 용어 추가/수정/삭제 기능 추가</li> <li>• 용어 출처 문장 표시 기능 추가</li> <li>• 용어 저장 기능 추가</li> <li>• 용어 검색 기능 추가</li> </ul>
--	--

코딩 결과 아래 그림 2의 웹 애플리케이션이 생성되었다. 전체적인 웹 애플리케이션 형태와 입력부 및 품사별 분류는 구현이 되었다. 그러나 품사별 추출이 원활하게 이루어지지 않아 추가 수정이 필요하였다. 미국 주식시장 관련 기사[5]를 입력한 결과, 명사 카테고리에는 대체로 전문용어 위주로 명사형 복합어가 추출된 편이나, ‘bringing it above’와 같은 분사구문이 포함되었고, 특히 동사 카테고리에는 지속적으로 ‘benchmark share index’, ‘closing level’, ‘downside surprise’ 등 명사 카테고리로 분류되어야 할 어휘들이 포함되고 추가 수정이 제대로 이루어지지 않는 문제가 발생하였다.

### 용어 정리 앱

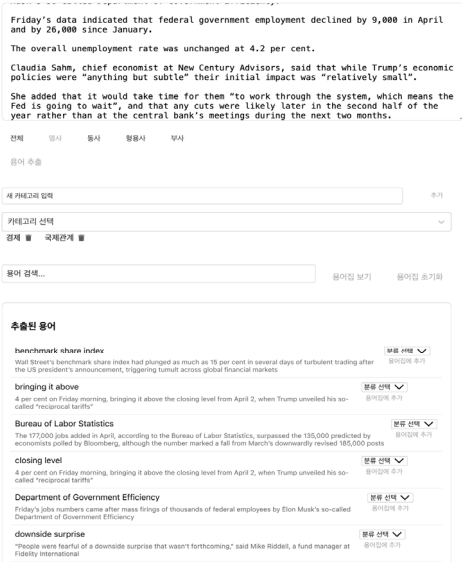


그림2. 1차 결과물

5) 입력 텍스트 출처: <https://www.ft.com/content/5ab8bce3-e3d9-48b8-a3a3-d8a563447ca8>

또한 카테고리 선택 및 분류, 용어 검색 등의 기능이 제대로 작동하지 않아 기능을 많이 추가하기보다는 최대한 기능을 단순화하는 방향으로 수정할 필요가 제기되었다. 추가 수정 사항은 3.2.2.절에서 설명하도록 하겠다.

### 3.2.2. 설계 수정

설계 수정 단계에서는 품사별 추출의 지속적 문제를 해결하기 위하여 분류체계를 단순화하였다. 전체, 전문용어, 일반표현으로 카테고리를 단순화하여, 전문용어 카테고리에는 최대 3 토큰의 명사형 복합어를 추출하고, 일반표현 카테고리에는 주제에 부합하는 유의미한 동사 위주로 최대 5 토큰의 복합어를 추출하도록 하였다. 그러나 앞선 프롬프트만으로는 기대하는 수준의 전문용어나 전문 동사구 추출이 원활하지 않았고 불완전한 구문이 다수 추출되었다. 이에 ChatGPT가 전문용어를 추출하는 기준을 참고하여 추가 프롬프트를 Cursor에 입력하였다. 한편, 기능 단순화를 위하여 카테고리 추가 및 용어 검색 기능은 삭제하였다. 대신 편의성을 위하여 전문용어집 저장 및 삭제 기능과 저장된 용어 목록을 엑셀 파일로 내보내는 기능을 추가하였다. 그리고 Google Translation API와 연동하여 추출된 용어를 한국어로 번역하는 기능을 추가하였다. 웹 애플리케이션은 로컬 호스트(<http://localhost:3004/>)에서 실행하였다. 주요 추가 내용 및 최종 구현 내용을 표 2에 정리하였다.

**표 2. 프롬프트 기반 수정 내용**

<p>프롬프트 수정 요청사항</p>	<ul style="list-style-type: none"> <li>• 카테고리 수정: 전체, 전문용어, 일반표현             <ul style="list-style-type: none"> <li>▪ 전문용어 추출 로직: 도메인 특화 용어, 행위가 아닌 대상 자체를 지칭하는 명사(구), 공식 기관명, 주변 문장과 별개로 독립적인 의미를 지닌 명사(구), 전문 약어, 글에서 반복되는 논지 연관 핵심 용어 포함; 일반 명사는 고정된 기술 용어의 일부일 때만 포함</li> <li>▪ 일반표현 추출 로직: 유의미한 동작/상태를 나타내는 동사 중심, 전치사/부사/목적어와 함께 고정된/반고정된 단위를 형성하는 동사구, 관용적 표현, 문맥 없이도 독립적인 의미를 가진 단위 포함</li> </ul> </li> <li>• 카테고리 추가 및 용어 검색 기능 삭제</li> <li>• 전문용어집 저장, 삭제, 엑셀 파일 내보내기 기능 추가</li> <li>• 추출된 용어별 대응 한국어 출력 기능</li> </ul>
<p>프롬프트 추가 수정 요</p>	<ul style="list-style-type: none"> <li>• 전문용어 추출 강화, 전문동사구 필터링 강화, 불완전 구문 필터링             <ul style="list-style-type: none"> <li>▪ 표제어(lemma)로 정규화</li> </ul> </li> </ul>

<p>청사향</p>	<ul style="list-style-type: none"> <li>▪ n-gram/구문 클러스터링: 유사도 높은 구문끼리 클러스터링</li> <li>▪ 불용어 및 약한 의미 표현 제거: 조동사, 부사 등</li> <li>▪ 낮은 빈도 구문 필터링</li> </ul>
<p>최종 구현구조</p>	<ul style="list-style-type: none"> <li>• 용어 추출 기능 <ul style="list-style-type: none"> <li>▪ 전문용어: 최대 3 토큰으로 제한, 전문용어 리스트와 매칭, 불필요한 패턴 필터링, 도메인 특화 단어 많은 구문 우선</li> <li>▪ 일반표현: 최대 5 토큰으로 제한, 동사 중심의 표현 추출, 전치사/부사와 결합된 동사구 포함, 전문동사구 우선</li> <li>▪ 불완전 구문 필터링 강화, 유사도 기반 클러스터링 임계값 상향 조정</li> </ul> </li> <li>• 번역 기능 <ul style="list-style-type: none"> <li>▪ 용어별 영한 번역 기능</li> <li>▪ 번역 버튼을 통한 개별 번역</li> <li>▪ Google Translation API 활용</li> <li>▪ 번역 결과 캐싱 처리</li> </ul> </li> <li>• 용어 관리 기능 <ul style="list-style-type: none"> <li>▪ 용어 저장/삭제 기능</li> <li>▪ 저장된 용어 카테고리별 분류</li> <li>▪ 원문과 번역 함께 표시</li> </ul> </li> <li>• 엑셀 내보내기 기능 <ul style="list-style-type: none"> <li>▪ 저장된 모든 용어 엑셀 파일로 내보내기</li> </ul> </li> </ul>

최종적으로 구현된 웹 애플리케이션 및 엑셀 저장 화면은 그림 3과 같다.

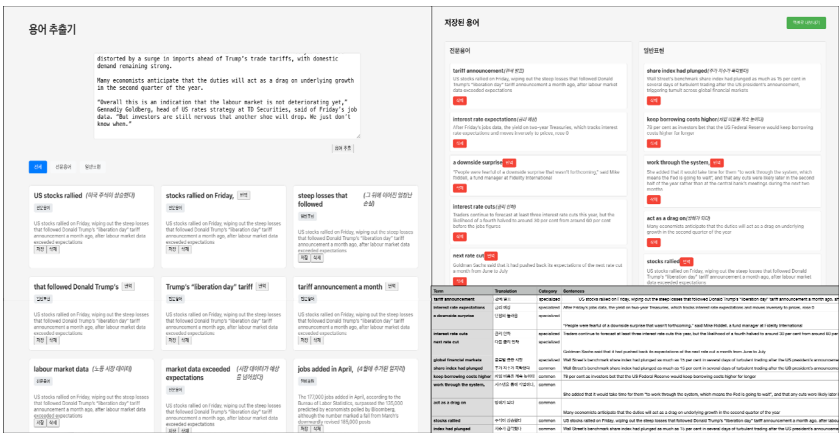


그림3. 용어 추출 웹 애플리케이션: 텍스트 입력부와 추출부(좌측), 사용자 저장 용어 목록(우측 상단), 엑셀 저장 파일(우측 하단)

최종적으로 구현한 용어 추출 웹 애플리케이션은 용어 추출, 용어 대응어 번역, 용어 저장, 엑셀 내보내기의 형식 기능 측면에서는 문제없이 구현되었다. 콘텐츠에 해당하는 용어 추출 측면에서도 연구자가 개인적으로 관심 가질만한 언어 조합이 적절한 수준으로 제시되어 개인 전용 앱으로 사용하기에 충분하다고 판단되나 프로토타입 수준 이상을 목표로 한다면 지속적 용어 추출 로직 조정을 통해 애플리케이션을 개선할 필요가 있다.

### 3.3. 통역 프로그램

최근 인공지능의 발달과 함께 생성형 인공지능 번역 기능이 발전을 거듭하고 있다. 이와 더불어 기계통역도 드물게나마 일부 회의 현장에서 시도되고 있어 인간 통역사들에게 막연한 위기감을 불러일으킬 수 있다(최선 2025). 그러나 실제 없는 불안감을 갖기보다는 해당 기술의 작동 원리와 구체적 장단점을 파악하는 노력이 전제되어야 할 것으로 판단하여 바이브코딩을 통해 기계통역 프로그램 프로토타입을 개발하였다.

일반적인 기계통역 혹은 자동통역 시스템의 원리는 다음과 같다. 음성인식 모듈이 언어1의 음성을 인식하여 문자열로 결과물을 생성하면, 자동번역기가 언어1의 문자열을 언어2의 문자열로 변환하고, 이후 응용분야에 따라 번역된 문자열을 자막으로 출력하거나 음성신호 합성을 거쳐 음성으로 출력한다(김승희 외 2014). 각 요소 기술은 자체개발을 하는 방식도 가능하겠지만 상용 서비스의 음성인식 및 합성 기능, 번역, Speech-to-Text, Text-to-Speech 등의 API를 연동하여 통합하는 방식을 채택할 수 있다(Oppenheimer 2024). 다만, 인공지능 번역 애플리케이션 대비 자동통역 기술 개발 선행연구에서 고질적인 문제로 거론되는 것은 음성 인식 품질과 전문용어 번역의 정확성 문제이다(김승희 외 2014; 장애리 2019; 최은실과 김아영 2020). 중국 인공지능 통역의 발전 현황을 분석한 장애리(2019)는 인공지능의 비교적 뛰어난 음성인식 수준에도 불구하고 한 명의 연사가 두 개 언어를 혼용하여 번역 엔진을 동시 가동시키는 경우, 연사가 무의미한 기능어를 단속적으로 반복함에도 번역엔진이 유의미한 값으로 변환하는 경우, 고유명사/약어/숫자 인식 오류가 발생하는 경우 등 음성 인식 문제와 전문용어 번역 오류 사례를 제시하였다. 한국 ETRI의 실시간 자동통역 시스템 사례를 분석한 최은실과 김아영(2020)도 전문용어 학습부족으로 인한 용어 자동번역 오류, 화자 방언 등으로 인한 음성인

식 오류, 구어체 인식 실패 오류, 자막표출 지연 문제, 문장 경계 인식 오류로 인한 부자연스러운 번역단위 분할 문제를 보고하였고, 이러한 문제들은 본고의 애플리케이션 개발 시 고려해야 할 사항들로 사료된다.

### 3.3.1. 설계

연구자는 영어와 한국어 간 동시통역 및 순차통역을 수행하고 원문과 통역 결과물을 텍스트로 시각화하여 제시하는 웹 애플리케이션을 설계하였다. 본 연구에서는 동시통역 기능에 집중하여 기능을 조정하고 개선하였다. 연구자의 요구사항에 따라 Cursor는 JavaScript와 Node.js 등을 이용하여 통역 웹 애플리케이션을 생성하였다. 통역 웹 애플리케이션은 ST(Source Text, 원문) 음성 인식 및 전사(Speech-to-Text, STT), 번역, TT(Target Text, 번역문) 시각화 기능이 핵심이며, 이 중 STT는 Web Speech API 또는 Google Cloud STT API와 연동하고, 번역 기능은 Google Translation API나 OpenAI API와 연동하면 애플리케이션을 구현할 수 있다. 본 연구에서는 STT 연동 API 종류별로 ‘통역 웹 애플리케이션 1’과 ‘통역 웹 애플리케이션 2’ 프로토타입 2종을 개발하였다. 통역 웹 애플리케이션 1에는 TTS(Text-to-Speech) 기능을 추가하여 통역 결과물의 음성화 기능 구현 가능성 여부도 검토하였다. 우선, 통역 웹 애플리케이션 1을 설명하고자 한다. 기본 구조는 표 3과 같다. 통역 웹 애플리케이션 1은 Safari 18.4와 Chrome 135.0.7049.115 브라우저를 통해 로컬 호스트(<http://localhost:3002/>)에서 실행하였다.

표 3. 통역 웹 애플리케이션 1에 사용된 주요 기술

연사 발화 청취 (음성 입력)	Web Speech API, socket.io: 브라우저에서 음성 인식, 서버로 전송
STT	Web Speech API: 음성→텍스트 변환
번역	OpenAI API, Google Translation API: 텍스트 번역
번역 시각화	socket.io: 번역결과 UI 표시
TTS	Web Speech API: 번역문 음성 합성
실시간 통신	socket.io: 실시간 데이터 송수신
서버/API 관리	Node.js, Express, dotenv: 서버, API 연동, 환경변수관리

특별한 미세조정 없이 구현한 결과 아래 그림 4 하단의 콘솔 기록에 표시된 바

www.kci.go.kr

와 같이 2~3 단어 단위로 통역 TT가 즉시 표출되어 지속적 문장 수정이 발생함에 따라 텍스트의 가독성이 현저히 저하되었다.

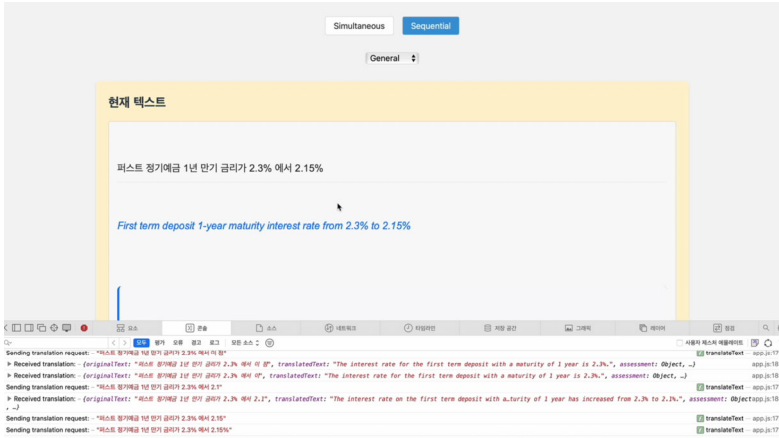


그림 4. 통역 웹 애플리케이션 1 초안-지속적 문장 수정 문제

이에 더해 애플리케이션이 특정 한국어 어휘에 대해 부적절한 번역을 계속 제시하는 문제가 있었다. 이에 용어집 기능을 추가하여 용어집에 정확한 대응어 쌍을 입력하면 번역에 반영되도록 하였다. 이상의 내용을 바탕으로 아래 표 4와 같이 프롬프트를 통해 수정 사항을 입력하였다.

표 4. 통역 웹 애플리케이션 1 구현 내용

추가 수정 프롬프트	<ul style="list-style-type: none"> <li>• 출발어와 도착어(통역 방향) 설정 버튼 추가</li> <li>• 문장 단위로 ST와 TT 시각화하여 가독성 개선</li> <li>• 새로운 문장이 기존 문장과 연결되어 번역 지연 발생하지 않도록 문장 단위 구분 강화 (번역 요청 타이밍 조정)</li> <li>• 순차통역에서 연사 발화 시에는 기다리다가 연사 발화가 끝나고 ‘통역해 주세요’라고 말하면 순차통역 시작</li> <li>• 영어 통역 결과물 TTS 시 원어인 발음 구현</li> <li>• 가독성 개선을 위해 ST와 TT를 문장 단위로 교차 출력</li> <li>• 용어집을 생성하고, 용어집에 입력한 국영문 대응쌍에 맞추어 해당 용어 번역</li> </ul>
구현 내용	<ul style="list-style-type: none"> <li>• 실시간 통역 기능</li> </ul>

	<ul style="list-style-type: none"> <li>▪ 음성인식(STT): 사용자의 음성을 실시간으로 텍스트로 변환(Web Speech API 사용)</li> <li>▪ 실시간 번역: 인식된 텍스트를 OpenAI API 또는 Google Translation API 를 연동하여 번역</li> <li>▪ 번역 결과 시각화: 문장 단위로 TT 를 실시간 표시(ST/TT 가 교차로 보이도록 UI 개선)</li> <li>▪ 음성 합성(TTS): TT 언어에 따라 한국어 또는 미국식/영국식 영어 등으로 읽어줌(Web Speech API 사용)</li> <li>• 용어집 관리 <ul style="list-style-type: none"> <li>▪ 엑셀 업로드: A 열 한국어, B 열 영어(또는 반대) 형식의 엑셀 파일 업로드로 용어집 자동 추가</li> <li>▪ 용어집에 입력한 단어/표현이 번역에 우선 반영되도록 구현</li> <li>▪ 용어집을 분야별로 관리, 새로운 분야 추가 가능</li> <li>▪ 용어집 자동 갱신: 업로드 후 클라이언트에서 용어집이 자동으로 갱신됨</li> </ul> </li> <li>• UI/UX 개선 <ul style="list-style-type: none"> <li>▪ 문장별 시각화: 번역 결과를 문장 단위로, ST/TT 가 교차로 보이도록 개선</li> <li>▪ 상단에 ST 실시간 STT 박스 유지</li> <li>▪ 하단에 TT 박스만 남기고, 문장별로 ST/TT 교차표시</li> </ul> </li> </ul>
--	--

STT 및 TTS는 브라우저에 내장된 무료 Web Speech API를 사용하였다. 번역 API는 당초 무료 OpenAI API(gpt-3.5-turbo-0125)를 1순위로, 유료 Google Translation API를 2순위로 설정하였는데, OpenAI API의 무료 한도가 조기에 소진되어 일정 기간 Google Translation API를 대안으로 사용하였다. 이후 OpenAI API도 유료로 전환하여 사용하였다. 표 4 내용을 바탕으로 구현된 실시간 통역 웹 애플리케이션 1 화면은 그림 5와 같다.

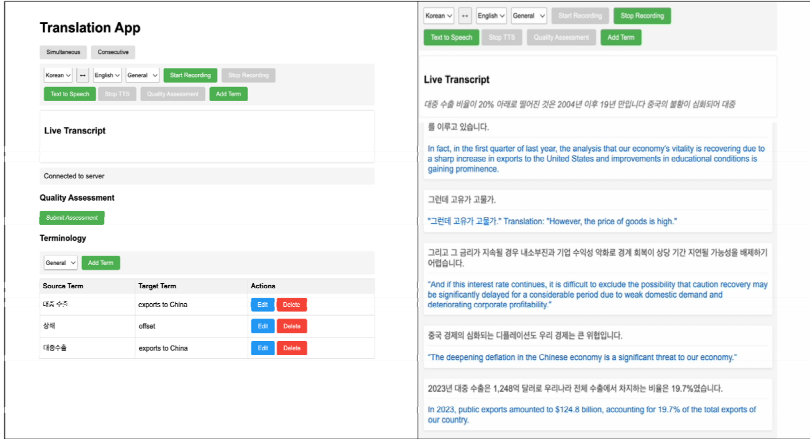


그림 5. 통역 웹 애플리케이션 1(Web Speech API 기반)

사용성 테스트 결과 무료 Web Speech API의 음성 인식 기능이 기대에 미치지 못하여 유료 Google Cloud Speech-to-Text API(이하, Google Cloud STT API)를 호출하는 새로운 통역 웹 애플리케이션 2 프로토타입을 개발하였다. 음성 인식 기능 차이를 비교하는 데 집중하고자 TTS나 용어집 등 부가적 기능은 생략하였다. 통역 웹 애플리케이션 2는 Safari 18.4와 Chrome 135.0.7049.115 브라우저를 통해 로컬 호스트(<http://localhost:3003/>)에서 실행하였으며, 구현 내용은 <표5>와 같다.

표 5. 통역 웹 애플리케이션 2 구현 내용

<p>프롭트</p>	<ul style="list-style-type: none"> <li>• 동시통역 웹 애플리케이션 개발 (동시통역 기능만 구현)</li> <li>• Google Cloud STT API 연동</li> <li>• 상단 박스에 실시간 발화 내용(ST) 표시(실시간 기록)</li> <li>• 하단 박스에 ST와 TT를 교차표시</li> <li>• 하단 박스에 번역 이력 유지, 새로운 세션 시작 시 초기화</li> <li>• 하단 박스에 ST와 TT는 상단 실시간 발화(ST) 기록 화면에 문장이 1~2문장 쌓이면 완성된 문장을 표시</li> </ul>
<p>구현 내용</p>	<ul style="list-style-type: none"> <li>• 실시간 통역 기능             <ul style="list-style-type: none"> <li>▪ 실시간 ST 수행, 완성된 문장 단위로 TT 제공</li> <li>▪ 마침표 등 문장 부호 추가</li> </ul> </li> <li>• 실시간 발화(ST) 박스             <ul style="list-style-type: none"> <li>▪ 사용자의 발화 내용(ST)을 실시간으로 보여줌</li> </ul> </li> <li>• 번역 결과물(TT) 박스</li> </ul>

	<ul style="list-style-type: none"> <li>▪ 완성된 문장만 추가, 각 문장 독립적 번역, 중복 번역 방지</li> <li>▪ 문장부호 없어도 일정 길이 이상이거나 문장부호가 중간에 있으면 번역 시도</li> <li>• 자동 에러 복구 및 UI/UX</li> <li>• 연동 API 목록 <ul style="list-style-type: none"> <li>▪ STT: Google Cloud STT API</li> <li>▪ 번역: Google Cloud Translation API</li> <li>▪ 클라이언트-서버 간 실시간 데이터 송수신: Socket.IO</li> <li>▪ Node.js 기반 웹 서버 프레임워크인 Express 사용</li> </ul> </li> </ul>
--	---

이상으로 구현된 통역 웹 애플리케이션 1, 2를 실행·분석한 결과는 동시통역 결과물을 중심으로 3.3.2.에서 소개하겠다.<sup>6)</sup>

### 3.3.2. 결과물 분석

#### (1) API 선택의 핵심성

3.2.절에서 검토한 용어 추출 웹 애플리케이션과 달리 통역 웹 애플리케이션은 기능별로 API를 통해 연동할 엔진·모델 선정이 핵심이다. 음성 인식과 발화 전사 단계에 해당하는 STT 단계는 Web Speech API 또는 Google Cloud STT API의 연동, 통역/번역 단계와 TT를 화면에 문자로 기록하는 단계는 OpenAI API 또는 Google Translation API와의 연동, 그리고 TTS 단계는 다시 Web Speech API를 연동하는 것이 기능 구현의 핵심이다. 따라서 개별 개발자가 코딩을 통해서 부가가치를 전달하는 범위는 인터페이스 계층에 해당하는 웹 애플리케이션의 레이아웃 구현과 통역 결과물인 TT가 효과적으로 표출되도록 최적의 텍스트 처리 방식을 찾는 것이다.

#### (2) 가독성- 문장 등 유의미한 의미 단위 구분 문제

앞서 3.3.1에서 설명하였듯, 최초 구현 시, 2~3 어휘 단위로 ST, TT 모두 지속적

---

6) 테스트용 국문 원문 출처:  
<https://www.futurekorea.co.kr/news/articleView.html?idxno=147221> (연구자 낭독)  
<https://youtu.be/bIGyylXkl0A?si=eLqiYle1fTV4Bz2-> (영상 음성 재생)

테스트용 영문 원문 출처:  
<https://www.ft.com/content/abf10a12-4bdc-4f58-aed1-9a445cae166b> (연구자 낭독)  
<https://youtu.be/5XCOWLjNKZA?si=qH0X3MVss8aXlJ1t> (영상 음성 재생)  
<https://www.youtube.com/watch?v=okmMvavpFes&t=1664s> (영상 음성 재생)

인 수정이 이루어짐에 따라 가독성이 현저히 저하되는 문제가 있었다. 통역 웹 애플리케이션 1과 2에서는 ST에서 완성된 1~2 문장 단위로 TT를 표출하도록 하여 가독성이 크게 향상되었다. 다만, 이 과정에서 엔진의 문장 인식 단위 파악이 수월하게 이루어지지 않아 TT 표출 시점이나 번역 품질에 영향을 미쳤다. 문장이 끝날 때마다 1~2초 간의 휴지를 두면 문장 구분이 정확히 되어 1~2 문장 단위별로 ST-TT가 화면에 적절히 표시되었다. 다만 의식적으로 문장 끝마다 휴지를 뒀아 하는 만큼 화자 입장에서 자연스러운 발화 흐름을 유지하기 어려웠다.



그림 6. 통역 웹 애플리케이션 2(Google Cloud STT API 기반)

휴지가 1초 이하인 연속 발화를 할 경우 실시간 발화 부분에만 ST가 계속 쌓이고 번역이 지연되거나 아예 진행되지 않는 문제도 발생하였다. 통역 웹 애플리케이션 2에서는 ST가 적체될 경우 완성된 문장은 TT로 먼저 표출되도록 수정하였다. 그러나 음성인식이 정확히 이루어지지 않아 ST가 비문으로 전사되면 애플리케이션 단에서 문장이 마무리되었다는 판단을 내리지 못하여 ST가 계속 적체되는 문제가 발생하였다. 이에 문장부호가 없더라도 ST가 적절한 길이 이상이 되면 번역을 시도하고 TT로 표출하도록 수정하였다. 그 결과 휴지 없는 빠른 발화 시에도 4~5 문장이 쌓이면 더 이상 적체시키지 않고 TT로 표출되었다(그림 6). 발화 속도

www.kci.go.kr

가 느리거나 구문 간 휴지가 1초 이상인 경우에는 해당 구문 단위로 번역이 이루어졌다. 이와 관련된 번역 품질 문제는 (4)에서 설명하겠다.

### (3) 음성 인식 문제

STT를 위해 연동한 무료 Web Speech API와 유료 Google Cloud STT API 간에는 확연한 음성인식 성능 차이가 관찰되었다. Web Speech API와 연동된 경우도 번역 모델은 LLM 기반이기 때문에 다소 부정확하게 인식된 음성도 맥락에 기반한 수정을 통해 번역 정확도를 높이는 경우도 있었다. 그러나 특정 어휘들은 아무리 반복적으로 정확하게 발음하여도 부정확하게 인식하는 경우도 발생하였다. 예를 들어, Safari 브라우저의 Web Speech API의 경우 한국어 출발어 ST, “중국의 불황이 심화되어 대중 수출이 급격히 하락한다면 최근 늘어난 대미 수출의 효과 상쇄를 불러온다”에서 한국어 ‘상쇄’가 문제였는데, 문장 전체의 맥락이 있음에도 불구하고 ‘상수’나 ‘상세’ 등으로 인식하였고, 문장 맥락에 따라 임의로 번역하여 문장이 전체적으로 왜곡되기도 하였다(그림 7).

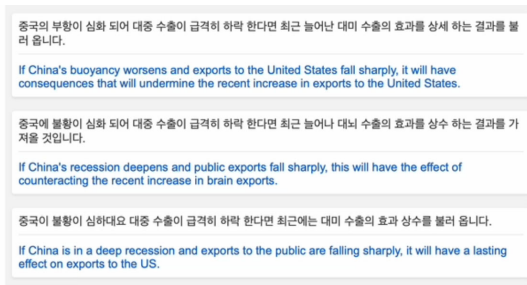


그림 7. 통역 웹 애플리케이션 1 결과물 - ‘상쇄’

같은 Web Speech API를 사용해도 브라우저에 따른 차이도 다소 있었다. 영어 ST, “Warren Buffett continued to sell stocks in the first three months of 2025 amid a broad market sell-off, according to a regulatory filing ahead of the historic 60th annual meeting of his holding company Berkshire Hathaway”에서 Safari는 ‘amid’를 제대로 인식하지 못하여 오역이 발생하였으나 Chrome에서는 제대로 인식하여 해당 부분은 비교적 적절히 번역되었다(그림 8).

Connected to server

Warren Buffett continued to sell stocks in the first three months of 2025 and made a broad market sell-off according to a regulatory filing ahead of the historic 60th annual meeting at this holding company Berkshire Hathaway.

규제 당국에 따르면, 워런 버핏은 2025년 1분기 동안 주식을 계속 매도했으며, 지주회사인 버크셔 해서웨이의 역사적인 60주년 연례 주주총회를 앞두고 광범위한 시장에서 매도를 단행했습니다.

Warren Buffett continued to sell stocks in the first three months of 2025 amended abroad market selloff according to a regular filing ahead of the historic 60th annual meeting at this holding company Berkshire Hathaway. The figures showing he offloaded stocks for the 10th consecutive quarter came in a quarterly earnings report that also noted Berkshire's insurance arm took a \$1.1 billion hit from wildfires in California in January.

워런 버핏은 2025년 첫 3개월 동안 해외 시장의 판매 압박에 따라 버크셔 해서웨이 소유 회사인 역사적인 60주년 연례 주주총회를 앞두고 정기 보고서에 따라 주식을 계속해서 매각했습니다. 10분기 연속으로 주식을 차분한 것을 보여주는 숫자는 버크셔의 분기 부문이 1월 캘리포니아 산불로 11억 달러의 손실을 입었음을 언급한 분기별 실적 보고서에 포함되었습니다.

사파리 브라우저, GPT-3.5-Turbo

Warren Buffett continue to sell stocks in the first 3 months of 2025 amid a broad Market sell-off according to a regulatory filing ahead of the historic 60th annual meeting of his holding company Berkshire Hathaway.

워런 버핏은 자신의 지분 회사 버크셔 해서웨이의 60주년 기념 행사를 앞두고 발표된 규제 청구서에 따르면 2025년 첫 3개월 동안 광범위한 시장 대각 속에서 주식을 계속해서 매도했습니다.

크롬 브라우저, GPT-3.5-Turbo

그림 8. 통역 웹 애플리케이션 1 결과물 - 'amid'

유틸리티 서비스인 Google Cloud STT API는 Web Speech API에 비해 전반적 품질이 나은 편이었지만 한국어 ST, “고유가, 고물가, 고금리가 지속되는 등”을 “그 육과 고물과고 금리”로 인식하고 “of the meat and the old and the high interest rates continue”로 번역하는 등 오류가 종종 발생하였다.

(4) 번역 품질 문제

기사문처럼 정형화된 문장을 정확하게 읽어 음성인식이 정확히 이루어진 경우, 각 번역 엔진이 기존 텍스트 번역에서 보여주는 성능을 대체로 발휘하는 편이었다. 그러나 위에서 기술하였듯, 음성인식 오류 발생 시 번역 품질 저하로 이어졌고, 번역 엔진 자체의 대응어 데이터 부족으로 인해 심각한 오역이 발생하기도 하였다.

어휘 수준에서는, 예를 들어, ‘대중 수출’이라는 표현을 Google Translation API를 통해 통역시켰을 때 줄곧 ‘public export’나 ‘mass export’로 통역하였다. 이를 보완하기 위하여 웹 애플리케이션 통역 시 정확한 대응어를 반영하도록 용어집에 ‘대중 수출’의 대응어로 ‘exports to China’를 입력하였으나 맥락에 따라 ‘exports to China’로 통역하다가도 다시 ‘mass export’를 혼용하는 문제가 지속되었다. OpenAI API 연동 시에는 문장 속에서는 ‘exports to China’로 통역하였으나 ‘대중 수출’ 단독으로 발화 시에는 ‘mass export’로 통역하였다. 이러한 유형의 오류는

보다 정교한 코드 수정이나 도메인 용어목록 보강 및 학습을 통해서 보완해야 할 것이다.

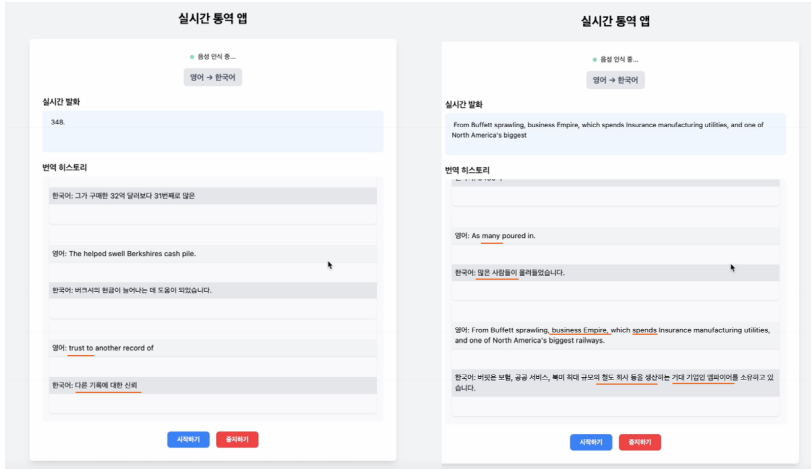


그림 9. 통역 웹 애플리케이션 2 - 비유창한 발화 인식 문제

또한 초분절적(suprasegmental) 층위에서 완벽한 하나의 문장 흐름으로 발화하지 않는 이상 일부 단어 뭉치가 독립적 의미 단위로 끊겨 의미 왜곡으로 이어지는 경우가 많았다. 그림 9에서 보듯, ST, “That helped swell Berkshire’s cash pile, / which rose to another record of / \$348bn / as money poured in / from Buffett’s sprawling business empire, which spans insurance, manufacturing, utilities and one of North America’s biggest railways”(휴지 부분 ‘/’ 표시)에서 약 2~3초간 휴지를 중간중간 주었을 때, 음성 인식, 분절 지점, 맥락 무시 오역 등 다수의 문제가 노출되었다. 이러한 문제들을 보완하기 위해 Cursor에서 한 가지를 조정하면 다른 유형의 발화에서는 품질이 역으로 저하되는 문제가 줄곧 발생하여 모든 경우를 만족하는 적절한 균형점을 찾는 것이 쉽지 않았다. 최상의 번역 결과를 원한다면 화자는 비유창성 요소 없이 깔끔하게 문장을 발화하고 문장 끝에는 휴지를 하도록, 즉, 기계에 자신의 발화 스타일을 맞추는 수밖에 없다.

## 4. 논의 및 결론

앞서 3장에서는 생성형 AI 기반 프로그래밍 지원도구를 이용한 바이브코딩 패러다임에 따라 통번역 관련 애플리케이션 프로토타입 개발 과정을 기술하였다. 4장에서는 이를 바탕으로 생성형 AI 시대 통번역사 실무 활용 가능성 및 통번역사 역할 확대 가능성을 논하고자 한다. 우선, 생성형 AI 프로그래밍 보조도구를 이용하면 요구사항을 프롬프트로 입력하는 것만으로도 간단한 애플리케이션을 생성할 수 있다. 이에 기존에는 코딩을 배우지 않으면 불가능의 영역으로 간주하였던 애플리케이션 개발이 가능해졌다. 전에는 상용제품에 의지해야 했던 다양한 기능들을 직접 개인 맞춤 개발하여 사용할 수 있는 1인 1 앱 시대가 열린 것이다. 본고에서는 필자에게 필요한 맞춤 용어 추출 애플리케이션을 만들어 필자의 언어 학습 니즈를 충족할 수 있었다. 마찬가지로 통번역사 누구나 바이브코딩으로 자신의 업무나 일상에 필요한 개인 맞춤 앱을 개발할 수 있다. 코딩을 전혀 몰라도 프로그래밍 보조도구가 알아서 앱을 개발해 줄 뿐 아니라, 그 과정에서 구현되는 기능과 연동된 코드를 참고하고 배울 수 있어 코딩 초보자의 학습 측면에서도 유용하다. 간단한 프롬프트 입력만으로도 코딩이 가능하다는 점은, 통번역 교육에서 프롬프트 엔지니어링을 단순히 텍스트 생성이나 교수법 개발에 국한하지 않고(신지선 2024; 진실희 2024), 보다 넓은 범주의 학습 역량으로 확장해 나갈 수 있는 가능성을 시사한다.

급속도로 발달하는 인공지능에 대한 막연한 불안감을 떨쳐내고 대비를 하는 데에도 도움이 된다. 비록 프로토타입 수준의 개발에 그치게 되더라도 애플리케이션의 작동 원리를 어느 정도 파악할 수 있어 막연한 불안감을 느끼기보다는 기술의 장단점과 보완이 필요한 부분을 파악하여 해당 부분에서 직업적 기회를 창출할 가능성도 모색해 볼 수 있다.

물론 바이브코딩의 한계도 분명하다. 코드 지식 없이 ChatGPT와 대화하듯 프롬프트로 생성형 AI에 일종의 ‘하칭’을 주는 것이기 때문에, 코딩 지식이 없다면 문제가 발생했을 때 실제로 코드의 어느 부분이 원인인지, 얼마나 잘못 작성되어 있는지를 알 길이 없다. 또한 프롬프트 대화를 바탕으로 AI가 한 가지 문제를 해결해 주더라도, 다른 문제를 초래하거나 기존에 정상적으로 작동하던 부분을 오히려 역행시키기도 한다. 따라서 프로토타입 수준을 넘어서는 개발을 생각한다면 전문 개발자와의 협업이 필수이다.

www.kci.go.kr

한편, 본 통역 웹 애플리케이션 프로토타입 개발 과정은 기계통역이 인간 통역사의 직업에 미칠 잠재적 위협 수준을 생각해 보는 기회가 되었다. 앞서 3장에서 기술하였듯 정형화된 문장을 정확하게 읽거나 발화하고 문장 간 휴지가 적절하게 들어간 경우의 성능은 양호하였다. 그러나 음성 인식 오류, 초본질 층위에서의 비정형적 발화, 다양한 발화 스타일 등으로 인해 STT 단계에서 오류가 발생하고, 이것이 이후 번역 단계에 영향을 미치는 경우가 관찰되었다. 또한 어휘 수준에서 오류가 종종 발생하였다. 오류 수준은 연동한 STT 및 번역 API에 따라 달랐으며, 성능이 더 좋은 엔진을 연동하더라도 부족한 부분은 용어집과 코드상으로 보완할 필요가 있었다. 최근 몇몇 방송국의 유튜브 뉴스 채널에서 사용한 기계통역인, 소위 ‘AI동시통역’에서도 비슷한 유형의 오류를 관찰할 수 있다(연합뉴스TV 2025; SBS뉴스 2025). 또한 최근 WHO 통역팀(WHO Interpretation Team, INT)이 기계통역 품질을 평가한 결과 평균 46%의 낮은 점수, 고유명사/숫자/문화적 지시어/전문용어 오역, 단조로운 TTS 음성, 통역 지연, 그리고 평판 리스크(정치·외교적 여파 또는 WHO 이미지 손상 등) 문제를 발견하였고, 이로 인해 대외 관계자와의 회의에서 기계통역을 사용하는 것이 부적합하다는 결론을 내린 바 있다(WHO INT 2025).

현재 기술로는 기계통역이 실력이 뛰어난 인간 통역사만큼 자연발화를 정확하게 통역하는 것은 어렵고, 특히 학습용 데이터베이스가 충분치 않은 수많은 전문도메인의 자연발화, 외교·문화적 함의가 담긴 발화, 미묘한 의미 차이가 담긴 발화를 정확히 통역하기는 어렵다. 기계통역 상용 서비스를 공급하는 플리토(Flitto)의 CEO도 한 유튜브 방송에 출연하여 기술 개발 전문가들도 기계가 전문성을 갖춘 인간 영역을 완전히 대체할 수 없다는 견해를 갖고 있다고 밝힌 바 있다(한국외국어대학교 통번역대학원 2024).

필자는 이에, 막연한 불안감을 갖기보다는 인간과 기술을 대립구도로 보는 인식을 극복하고 AI가 창출하는 새로운 기회에 주목하고자 한다. 기계통역은 정확성이 중요한 현장에서 사용하기는 어렵겠지만, 기존에 통번역 제공을 전혀 고려하지 않던 현장에 적용될 수 있을 것이다. 이때 기술적으로 부족한 기계통역을 완성하는 것은 결국 언어 전문가인 통역사의 몫이다. 도메인별 전문용어 관리, AI가 인식해야 할 적절한 의미 단위 분석, 다양한 ST-TT 쌍 생성 및 관리, 기계 통역 결과물의 품질 평가 및 관리 등이다. 언어 전문가이자 도메인 전문가인 전문 통역사들이 각자의 현장에서 기술이 필요한 경우는 언제이고 사람이 필요한 경우는 언제인지,

기술이 필요하다면 다양한 옵션 중에서 어떤 것을 선택할지를 조언하는 컨설턴트이자 최적의 선택을 판단하는 결정자가 되어야 한다. 프리랜서 통역사로 국제회의 행사에 관여하건, 인하우스 통역사로 사내의 통번역 관련 기술에 관여하건, 통역사가 전문가로서 가장 잘 해낼 수 있는 책임자이기 때문이다. 그리고 그러한 판단을 내리기 위해서는 언어 및 통번역 전문성뿐 아니라 기술에 대한 이해도 필요하다. 바이브코딩은 이러한 용도로 효과적으로 활용될 수 있다. 더 나아가 기술 이해도를 바탕으로 언어 및 통번역 관련 기술 공급 사업자로 통번역사의 역할을 확대할 수도 있을 것이다.

다만, 인공지능의 발달 과정에서 우리가 간과하지 말아야 할 것이 있다. 아무리 기계의 번역 품질이 급격히 향상된다 해도 최종 결과물의 품질을 검수할 인간 통번역사의 역할은 여전히 필요할 것이다. Cursor가 제아무리 자동으로 코딩을 해준다 해도 코드를 모르는 이용자는 오류를 발견하지도, 수정하지도 못한다. 언어도 마찬가지이다. 기계 통번역 결과물의 신뢰성 여부를 판단하는 최종 검수자는 양쪽 언어에 능통하고 주제지식도 갖춘 전문 통번역사의 역할이 될 것이다.

발화 메시지의 발신자만이 아닌 수신자인 청자와 독자의 입장도 고려해야 한다. 낮은 품질의 결과물을 일방적으로 제공하고 만다면 청자와의 소통은 요원해진다. 기계통역이 자막으로만 제공되는 경우 청자는 자막을 읽느라 연사의 목소리 이외의 표정이나 제스처 등 시각 정보를 놓치고 만다. 이는 소통 당사자 중 일방을 약자로 내모는 윤리적 문제를 야기한다. 기계통역 방식에 맞춰 연사가 평소의 발화 스타일을 포기하고 기계 스타일에 자신을 맞추기 시작하면 발화 스타일의 독창성과 다양성이 사라진다. 결국 언어 소통의 본질이 무엇인지를 재차 환기할 필요가 있다. 언어에는 텍스트만이 아닌 해당 언어권의 문화와 가치관이 담겨 있다. 그리고 소통은 말로만 하는 것이 아니라 발화자와 청자의 오감을 통해서 맥락과 배경 지식을 바탕으로 진행된다. 통역 사용자를 대상으로 인간 통역, AI 통역, 기계통역 자막 중 선호하는 모드를 조사한 배문정(2023)의 연구에서도 사용자들이 편안한 음성, 자연스러운 어조, 연사의 강조점 반영, 직원들과 자유로운 회의 주제 논의, 조직 내부 지식 보유, 돌발 사태 대처 등 언어 외적 측면 때문에 인간 통역사를 선호한다는 점이 강조된 바 있다. 이미 접어든 AI 시대에, 소통의 본질을 잃지 않으면서 기술 활용도를 높이는 적절한 균형점을 찾고 기계와 인간이 공존하는 세상을 만들어 나가길 기대하는 바이다.

## 참고문헌

- 김승희·박준·김상훈(2014). 「자동통역기술, 서비스 및 기업 동향」. 『전자통신동향분석』 29(4): 39-48.
- 배문정. (2023). 「통역 사용자를 대상으로 한 통역 모드 선호도 조사: 인간 통역, AI 통역, 자막 비교」. 『번역학연구』 24(3): 592-614.
- 신지선. (2024). 「번역교육에서 챗GPT의 활용 방안 탐구」. 『번역학연구』 25(2): 39-67.
- 이미경 (2018). 「전문 용어의 사용이 통역의 과정에 미치는 영향에 대하여」. 『통번역교육연구』 16(3): 127-146.
- 이준호·김순미. (2022). 「풀 포스트에디팅에 대한 고찰—풀 포스트에디팅 생산성에 영향을 주는 요소를 중심으로」. 『번역학연구』 23(5): 119-146.
- 장애리. (2019). 「중국의 인공지능(AI) 통역 발전 현황 분석」. 『번역학연구』 20(5): 163-195.
- 진실희. (2024). 「통역 교육 콘텐츠 개발을 위한 챗GPT 프롬프트 설계: 제작과 편집을 중심으로」. 『번역학연구』 25(3): 123-150.
- 최문선. (2018). 「전문통역사의 용어집 구축에 관한 예비연구」. 『통번역학연구』 22(2): 179-206.
- 최문선. (2019). 「전문분야 통역을 위한 어휘 지식: IT 분야를 중심으로」. 『번역학연구』 20(3): 247-283.
- 최은실·김아영. (2020). 「한-영 실시간 자동통역 시연 사례 고찰」. 『통번역교육연구』 18(2): 125-147.
- 허지운. (2016). 「언론보도에 투영된 동시통역에 대한 고찰」. 『통역과 번역』 18(2), 153-187.
- Barke, S., James, M. B. and Polikarpova, N. (2023). Grounded copilot: How programmers interact with code-generating models. In *Proceedings of the ACM on Programming Languages*, 7(OOPSLA1), *ACM Journals*, 85-111.
- Briggs, J. and Kodnani, D. (2023). The Potentially Large Effects of Artificial Intelligence on Economic Growth.  
<https://www.gspublishing.com/content/research/en/reports/2023/03/27/d64e052b-0f6e-45d7-967b-d7be35fabd16.pdf>(2025.3.2. 검색).
- Davila, N., Wiese, I., Steinmacher, I., Lucio da Silva, L., Kawamoto, A., Favaro, G. J. P. and Nunes, I. (2024). An industry case study on adoption of ai-based programming assistants. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Practice*. *ACM*, 92-102.
- Fakhoury, S., Naik, A., Sakkas, G., Chakraborty, S. and Lahiri, S. K. (2024). LLM-based test-driven interactive code generation: User study and empirical evaluation. *IEEE Transactions on Software Engineering* 50(9): 2254-2268.
- Fan, G., Liu, D., Zhang, R. and Pan, L. (2025). The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: A comparative study with traditional pair programming and individual approaches. *International Journal of STEM Education* 12: 16.
- Fantinuoli, C. (2017). Computer-assisted preparation in conference interpreting. *Translation &*

- Interpreting* 9(2): 24-37.
- Sajja, R., Ramirez, C. E., Li, Z., Demiray, B. Z., Sernet, Y. and Demir, I. (2024). Integrating Generative AI in Hackathons: Opportunities, Challenges, and Educational Implications. *Big Data and Cognitive Computing* 8: 188.
- Sun, D., Boudouaia, A., Zhu, C. and Li, Y. (2024). Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study. *International Journal of Educational Technology in Higher Education* 21: 14.
- Vaithilingam, P., Zhang, T. and Glassman, E. L. (2022). Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *Chi Conference on Human Factors in Computing Systems Extended Abstracts*. ACM, 1-7.
- Weisz, J. D., Kumar, S., Muller, M., Browne, K. E., Goldberg, A., Heintze, E. and Bajpai, S. (2025). Examining the use and impact of an AI code assistant on developer productivity and experience in the enterprise. [https://arxiv.org/pdf/2412.06603\(2025.4.1. 검색\)](https://arxiv.org/pdf/2412.06603(2025.4.1. 검색)).
- White, J. (2024). Building living software systems with generative & agentic AI. [https://doi.org/10.48550/arXiv.2408.01768\(2025.4.2. 검색\)](https://doi.org/10.48550/arXiv.2408.01768(2025.4.2. 검색)).
- WHO INT. (2025). Report on WORDLY AI Interpretation. [https://www.linkedin.com/posts/ghada-chadarevian-444966230\\_who-report-on-ai-interpretation-activity-7333843691989573633-Z5L?utm\\_source=share&utm\\_medium=member\\_desktop&rcm=ACoAACQIIQgB5HFHq91FTmeeLew-8nv1J09eRN4\(2025.6.7. 검색\)](https://www.linkedin.com/posts/ghada-chadarevian-444966230_who-report-on-ai-interpretation-activity-7333843691989573633-Z5L?utm_source=share&utm_medium=member_desktop&rcm=ACoAACQIIQgB5HFHq91FTmeeLew-8nv1J09eRN4(2025.6.7. 검색)).

[인터넷 자료]

- 연합뉴스TV. (2025). [풀영상/실시간 번역] 트럼프, 광물협정 임박 시사... "첼렌스키, 중요한 서한 보내와" 미 연방 의회 연설 풀영상/연합뉴스TV(YonhapnewsTV). [https://www.youtube.com/live/2566xKLj6Qw\(2025.3.5.검색\)](https://www.youtube.com/live/2566xKLj6Qw(2025.3.5.검색)).
- 이종현. (2023). "구글의 시대 끝났다" 평가 나온 챗GPT... AI 대화가 검색 대체할까. [https://biz.chosun.com/science-chosun/science/2022/12/06/3GXWJJI6WNGCPIJGBLBDQCEA6OI/?utm\\_source=chatgpt.com\(2025.3.2.검색\)](https://biz.chosun.com/science-chosun/science/2022/12/06/3GXWJJI6WNGCPIJGBLBDQCEA6OI/?utm_source=chatgpt.com(2025.3.2.검색)).
- 최선. (2025). 의학회도 본격적인 AI 시대...실시간 강의 번역 기본. [https://www.medicaltimes.com/Main/News/NewsView.html?ID=1163338\(2025.5.6.검색\)](https://www.medicaltimes.com/Main/News/NewsView.html?ID=1163338(2025.5.6.검색)).
- 한국외국어대학교 통번역대학원. (2024). 플리토 대표에게 묻다! AI 통번역의 미래는? 플리토 CEO 이정수 인터뷰. [https://www.youtube.com/watch?v=JnQXfdNuuRA\(2025.4.28.검색\)](https://www.youtube.com/watch?v=JnQXfdNuuRA(2025.4.28.검색)).
- SBS뉴스. (2025). AI동시통역 트럼프 에어포스원 긴급 기자회견 "내일 푸틴과 대화"..우크라이나 전 '급물살' SBS 실시간 라이브. [https://www.youtube.com/watch?v=T5fyJSDF5xE&t=683s\(2025.3.17.검색\)](https://www.youtube.com/watch?v=T5fyJSDF5xE&t=683s(2025.3.17.검색)).
- Asplund, J-E. (2025). Cursor at \$100M ARR. [https://sacra.com/research/cursor-at-100m-arr/\(2025.2.10.검색\)](https://sacra.com/research/cursor-at-100m-arr/(2025.2.10.검색)).
- Edwards, B. (2023). OpenAI's GPT-4 exhibits "human-level performance" on professional benchmarks.

- <https://arstechnica.com/information-technology/2023/03/openai-announces-gpt-4-its-next-generation-ai-language-model/>(2025.4.2. 검색).
- Harkar, S. (2025). What is vibe coding?.  
<https://www.ibm.com/think/topics/vibe-coding#:~:text=Vibe%20coding%20is%20a%20fresh,1>(2025.4.9. 검색).
- IBM. (n.d.). Machine-aided translation.  
<https://www.ibm.com/history/machine-aided-translation>(2025.4.2. 검색).
- Karpathy, A. (2023). <https://x.com/karpathy/status/1617979122625712128>(2025.3.3. 검색).
- Karpathy, A. (2025). <https://x.com/karpathy/status/1886192184808149383>(2025.4.2. 검색).
- Kelly, J. (2025). AI and automation: Threats and opportunities for white-collar professionals.  
[https://www.forbes.com/sites/jackkelly/2025/03/02/ai-and-automation-threats-and-opportunities-for-white-collar-professionals/?utm\\_source=chatgpt.com](https://www.forbes.com/sites/jackkelly/2025/03/02/ai-and-automation-threats-and-opportunities-for-white-collar-professionals/?utm_source=chatgpt.com)(2025.5.3. 검색).
- Lopez, J. (2023). <https://x.com/jvilopen/status/1719363262179938401>(2025.3.3. 검색).
- McNulty, N. (2025). How I vibe-coded a micro app in 10 minutes with ChatGPT.  
<https://medium.com/@niall.mcnulty/how-i-vibe-coded-a-micro-app-in-10-minutes-with-chatgpt-87db79fe5b4a#:~:text=Of%20course%2C%20the%20initial%20script,more%20customised%20to%20my%20setup>(2025.4.5. 검색).
- Oppenheimer, J. (2024). How to build a translator bot using web speech API and GPT-3.  
<https://webtc.ventures/2024/03/how-to-build-a-translator-bot-using-web-speech-api-and-gpt-3/>(2025.4.5. 검색).
- Pangeanic. (2023). What is the size of the translation industry?.  
<https://blog.pangeanic.com/what-is-the-size-of-the-translation-industry>(2025.4.2. 검색).
- Shani, I. (2023). Survey reveals AI's impact on the developer experience.  
<https://github.blog/2023-06-13-survey-reveals-ais-impact-on-the-developer-experience/>(2025.3.2. 검색).
- Smith, M. S. (2025a). Engineers are using vibe coding to code based on vibes.  
<https://spectrum.ieee.org/vibe-coding#:~:text=While%20Vohra's%20take%20on%20vibe,stacks%20they%20don't%20yet%20understand>(2025.4.9. 검색).
- Smith, M. S. (2025b). I started 'vibe coding' my own apps with AI. I'm absolutely loving it.  
<https://www.pcworld.com/article/2660539/i-started-vibe-coding-my-own-apps-with-ai-im-absolutely-loving-it.html>(2025.4.20. 검색).
- Stackoverflow. (2023). 2023 Developer Survey StackOverflow.  
<https://survey.stackoverflow.co/2023/#developer-tools-ai-tool-interested>(2025.3.2. 검색).
- Tap Twice Digital. (2025). 10 Cursor statistics: Revenue, valuation, competitors, founder.  
[https://taptwicedigital.com/stats/cursor?utm\\_source=chatgpt.com](https://taptwicedigital.com/stats/cursor?utm_source=chatgpt.com)(2025.5.1. 검색).
- Wiggers, K., Corral, C. , Stringer, A. and Park, K. (2025). ChatGPT: Everything you need to know about the AI-powered chatbot.  
<https://techcrunch.com/2025/05/01/chatgpt-everything-to-know-about-the-ai-chatbot/>(2025.5.23. 검색).

This paper was received on 9 May 2025; revised on 9 June 2025; and accepted on 16 June 2025.

---

*Authors' email address*

huhjiun@ewha.ac.kr

*About the author*

Jiun Huh is an Associate Professor at the Graduate School of Translation and Interpretation (GSTI) at Ewha Womans University. Her research interests include interpretation technologies, market trends, and interpreter training.

www.kci.go.kr

