초월번역기를 만들어 보자

데이터-AI 빌리지
최민주

발표자 소개

번역기 만들려고 커리어 바꿔서 AI 함

- 이름 : 최민주 (Judy Choi)
- 직업 : AI Engineer

- 기계번역 관련 경력
  - 법률 전문 번역 스타트업 근무 (2020 ~ 2021)
  - 국내 학회 기계번역 논문 4편 발표

- 지금은 안해요?
  - 네….

개발자가 AI로 대체될까 봐 걱정을

한다 VS 안 한다

개발자가 AI로 대체될까 봐 걱정을

한다 VS 안 한다

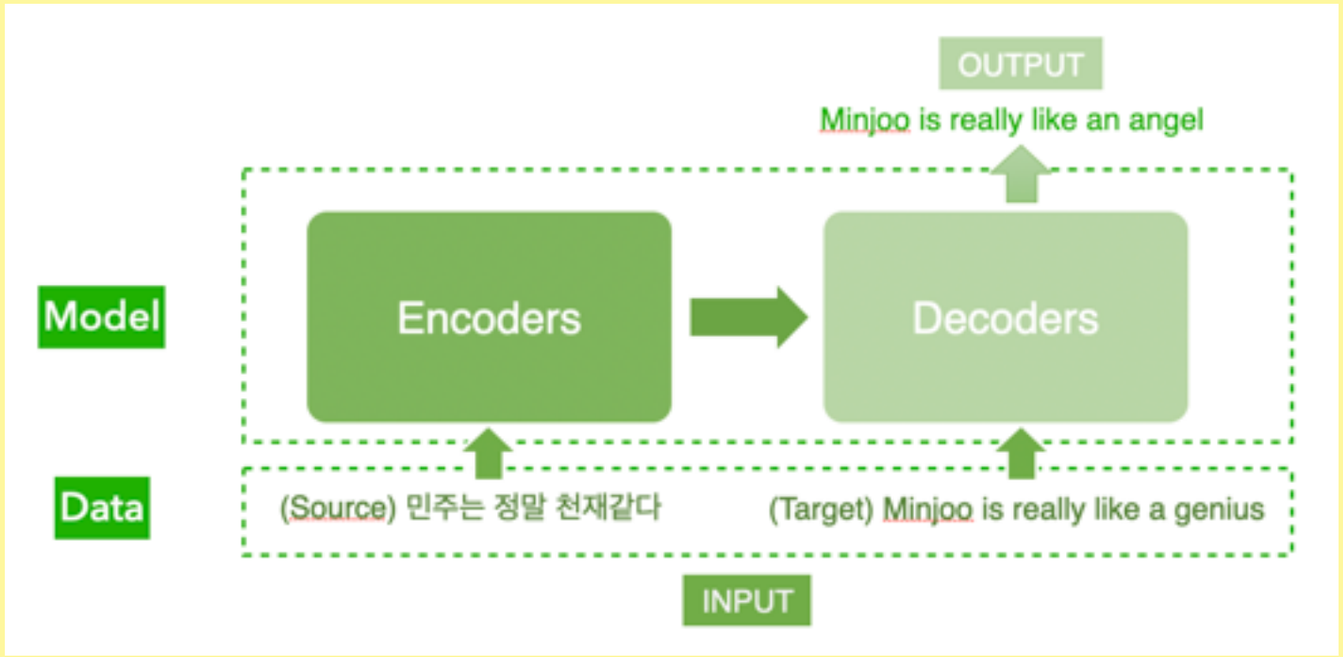

역시 형이야
구하러 왔구나
아니 나도 잡혔어

이미 대체됐어요.. 😇

ChatGPT 등장 전의 기계번역

- 기계번역기 개발 순서
    - 언어별 번역 데이터셋 구축 ➡️ 모델 학습 ➡️ 품질 평가

- 언어별 데이터셋 구축
    - 데이터 수집
    - 데이터 정제
- 모델 학습
    - 값비싼 GPU
    - 긴 학습 시간 * 반복 실험



이미 대체됐어요.. 😇

ChatGPT 등장 후의 기계번역

- 기계번역기 개발 순서
  - 프롬프트 작성 ➡️ ChatGPT(LLM API) 호출 ➡️ 품질 평가

- 프롬프트 작성
  - AI 지식이 없어도 가능
- ChatGPT(LLM API) 호출
  - 코드 몇 줄로 해결

```python
import os
from openai import OpenAI

response = OpenAI(api_key=os.getenv("OPENAI_API_KEY")).chat.completions.create(
    model="gpt-4-turbo",
    messages=[
        {"role": "system", "content": "주어진 문장을 영어로 번역하세요."},
        {"role": "user", "content": "나는 남자 보는 눈이 높아요"}
    ]
)

translated = response.choices[0].message.content
```

여전히 AI가 넘지 못한 언어의 장벽

가장 인간적인 장르 : 문학

- 문학 작품(literary texts) 번역
    - 언어 자체의 복잡성, 묘사적 표현, 문화적인 뉘앙스 등등

- 예시) '채식주의자' - 한강

원문	향긋하고 달콤하게 튀긴 삼겹살
직역	Fragrant and sweet fried pork belly
번역	Fragrant, caramelized deep-fried belly pork


Agent

AI Agent?

- 특정 작업이나 목표를 수행하기 위해 설계된 인공지능 시스템
- 인간의 개입 없이 스스로 문제를 해결하고, 판단을 내리고, 목표를 달성할 수 있는 소프트웨어

- 예시
  - 고객 서비스 등 사용자의 질문에 대해 적절한 답을 제공하는 **챗봇**
  - 도로 환경을 인식하고, 자율적으로 주행 결정을 내리는 **자율주행차**
  - 알람 설정, 날씨 정보 제공 등 사용자가 원하는 작업을 수행하는 **AI 비서**

AI Agent?
구성 요소

- **Task**
  - 수행해야 할 구체적인 과제나 작업
    - 특정 주제에 대한 블로그 포스트 작성하기

- **Agent**
  - 작업을 수행하는 주체
    - 자료 검색 Agent
    - 글 작성 Agent

- **Tool**
  - Agent 가 Task를 수행하는 데 사용하는 도구
    - 웹사이트 스크랩 API
    - 웹사이트 콘텐츠 검색 API

- **Orchestration Layer / Process / Router**
  - Agent 들이 태스크를 효율적으로 수행하도록 작업의 흐름을 조정
  - 여러 모델, 도구, 또는 파이프라인을 관리하고 조율

TransAgent

LLM을 활용한 Multi-Agent 시스템

- 가상의 번역 회사 'TransAgent' 설립
  - 번역 회사의 직원을 각각의 Agent로 구현
    - CEO
    - 선임 편집자
    - 주니어 편집자
    - 번역가
    - 현지화 전문가
    - 교정자

Figure 2: TRANSAGENTS, a multi-agent virtual company for literary translation.

✳ Agent : 특정 작업을 수행하는 자율적인 단위 또는 개체

(Perhaps) Beyond Human Translation: Harnessing Multi-Agent Collaboration for Translating Ultra-Long Literary Texts

TransAgent

LLM을 활용한 Multi-Agent 시스템

- 개별 에이전트에게 역할을 부여하기 위해서 GPT-4-Turbo 모델과 시스템 프롬프트 사용
- 개별 역할들에 대한 가상의 에이전트 프로필 (virtual agent profiles)을 30개 생성

```
Name: Sofia Chang
Languages: English, Mandarin, Spanish, French
Nationality: Canadian
Gender: Female
Age: 47
Education: Ph.D. in Comparative Literature
Personality: meticulous, introverted,
↪  perfectionist, critical, thoughtful
Hobbies: gardening, chess, watercolor painting
Rate per word: 0.12
Years of working: 22
Profession: Senior Editor
Role prompt: You are Sofia Chang, a highly esteemed
↪  Senior Editor [TRUNCATED]
```

Figure 3: An example profile of **Senior Editor**.

- 작업 순서
  - CEO 에이전트가 작업에 맞는 선임 편집자 선택
  - 선정된 선임 편집자가 팀원을 모아 팀 구축
  - 팀원들은 협업하여 번역 작업 수행

참고 : https://hayunjong83.tistory.com/63

TransAgent

LLM을 활용한 Multi-Agent 시스템

- 결과
    - TransAgent 는 d-BLEU scores에서는 일관되게 가장 나쁜 성능 (=번역 성능 지표의 점수는 낮음)
    - 그렇지만 사람이 한 번역(human-written reference translation)과 GPT-4 번역과 비교할 때,
        - 인간 평가자와 LLM 평가 모두 TransAgent 의 번역 결과를 선호하였다.
        - 구체적으로 역사적인 문맥이나 문화적인 뉘앙스와 같은 도메인 지식이 요구되는 장르에서는 사람의 번역보다 뛰어났다.
        - 그러나 현시대의 상황이 반영된 장르(contemporary genre)에서는 성능이 떨어졌다.
- 문학 번역에 소요되는 TransAgent 비용은 전문 번역가를 고용하는 것보다 80배 저렴

참고 : https://hayunjong83.tistory.com/63


야너두

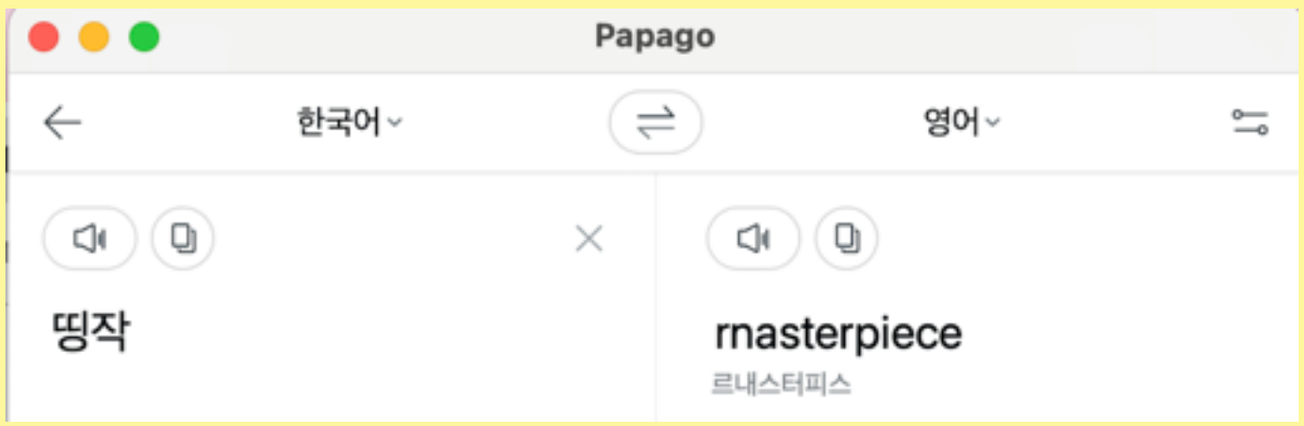
실험 목표 / 조건

- 단순 LLM 을 이용하는 번역의 한계를 극복해 보자
- 이왕이면 트렌디한 기술을 써 보자 ➡️ AI Agent 를 적용해 보자
- TransAgent 는 혼자 구현하기에 규모가 크다 ➡️ 먼저 작은 Agent 를 구현해 보자

여전히 AI가 넘지 못한 언어의 장벽

창의적인 장르 : 초월번역

- 초월번역?
  - 원문의 의미, 느낌을 직역한 것보다 더 효과적으로 표현한 번역을 뜻하는 말로 창작 이상의 창의적 번역 작품을 만났을 때 쓰는 표현
  - 외국어 원문의 뜻을 훼손하지 않고 현지 문화권의 느낌이 살아나도록 의역한 경우를 칭찬하는 표현



아시아경제 (https://www.asiae.co.kr/article/20190621192633657 56)

김경훈 한국트렌드연구소 소장 (https://www.hani.co.kr/arti/specialsection/esc_section/885838.html)

초월번역기를 만들어 보자

초월번역 예시 문장

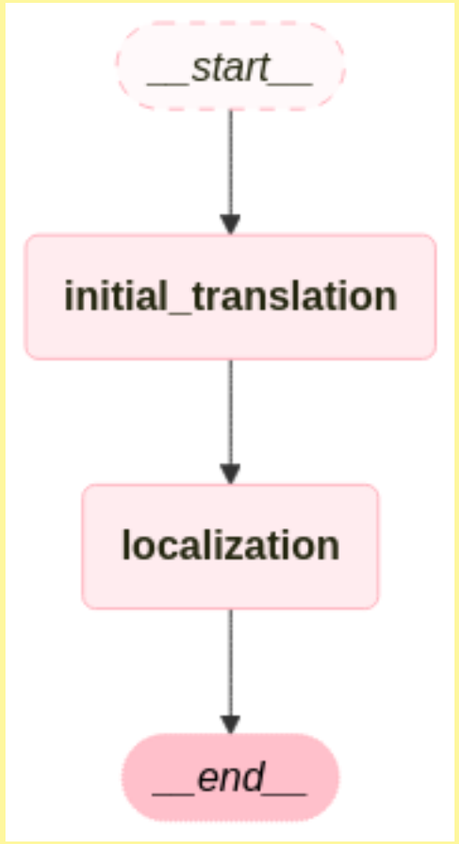
원문	What were you guys smokin’ when you came up with that?
직역	그걸 생각해냈을 때 여러분은 무엇을 피우고 있었나요?
초월번역	

초월번역기를 만들어 보자

초월번역 AI Agent 설계

- 초벌 번역가 (initial_translator) Agent
  - 주어진 영어 문장을 한국어로 정확하게 번역하는 콘텐츠 전문 번역가
  - 의역이나 창의적 번역을 하지 않도록 설정

- 현지화 전문가 (localizer) Agent
  - 번역된 콘텐츠가 한국 문화와 언어에 맞게 자연스럽고 이해하기 쉽게 수정
  - 창의성을 발휘하도록 설정

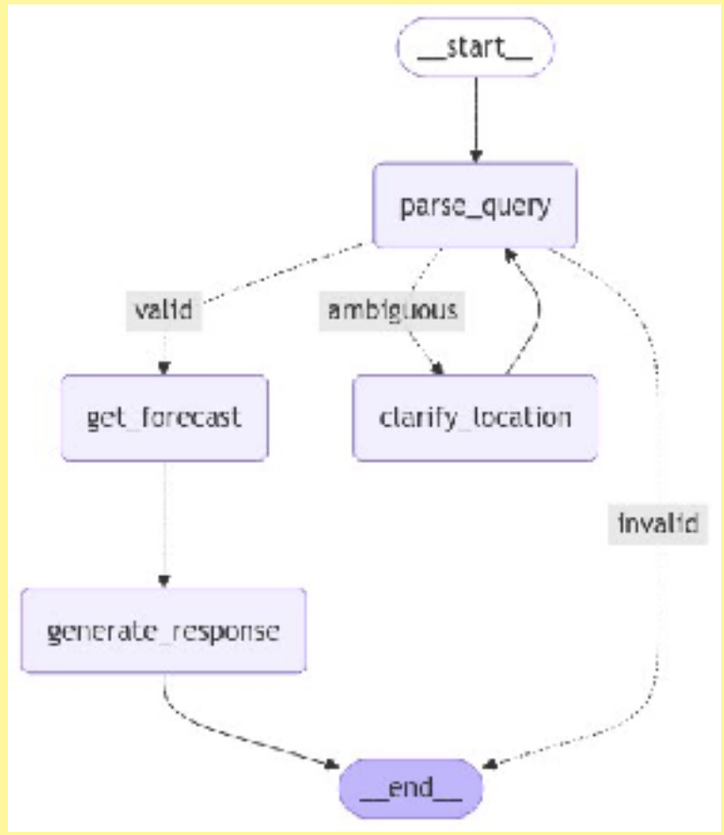


초월번역기를 만들어 보자

LangGraph로 만들어 보자

- 자연어처리 및 AI 응용 프로그램 개발을 위한 프레임워크

- 장점
  - 복잡한 작업을 그래프 형태로 쉽게 구현하고 관리할 수 있음
  - 각각의 독립된 모듈 단위로 개발 (레고 블록 쌓듯이)
  - LangSmith 를 이용해 모니터링 가능

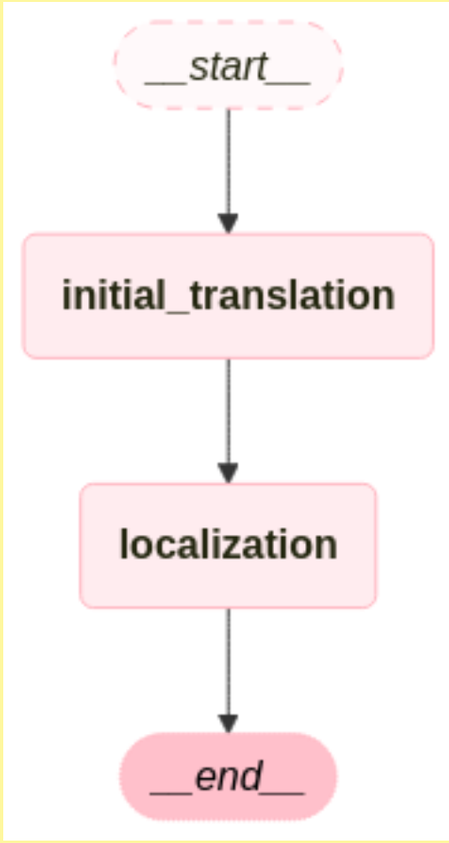


예시 : https://wikidocs.net/261598

LangGraph
구성 요소

State

요소	정의
State	현재 상태를 나타내는 공유 데이터 구조
Node	실제 작업을 수행하는 함수
Edge	노드 간 연결



참고 : https://wikidocs.net/261590

LangGraph

구현 예시 : State 정의 / Graph 인스턴스 생성

1. State 정의

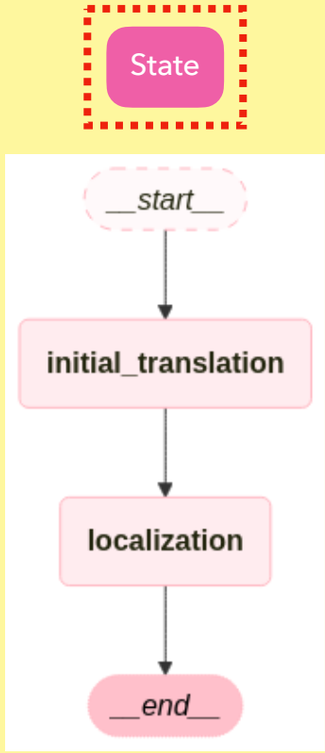
```python
from typing import TypedDict

class State(TypedDict):
    text: str
```

2. Graph 인스턴스 생성

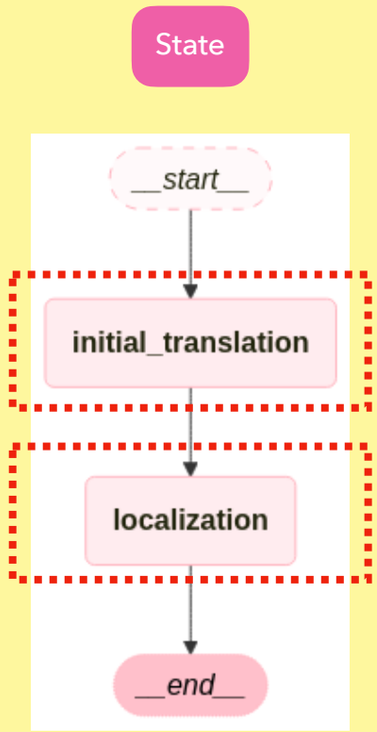
```python
from langgraph.graph import StateGraph

# LangGraph 그래프 생성
graph = StateGraph(State)
```



LangGraph

구현 예시 : Node 정의 & 추가



3. Node 정의

```python
# Node 정의
from langchain.schema import SystemMessage, HumanMessage

# 첫 번째 노드: 초벌 번역 (영어 → 한국어)
def initial_translation(state: State):
    prompt = [
        SystemMessage(content="당신은 콘텐츠 전문 번역가입니다. 사용자가 제공하는 문장을 한국어로 번역하세요."),
        HumanMessage(content=state["text"])  # ✅ State 의 'text' 키 사용
    ]
    translated_text = initial_translation_llm(prompt).content
    return {"text": translated_text}  # ✅ 딕셔너리 형태로 반환

# 두 번째 노드: 현지화 (자연스러운 한국어로 수정)
def localization(state: State):
    prompt = [
        SystemMessage(content="""
                      당신은 한국어 콘텐츠 현지화 전문가입니다.
                      초벌 번역된 한국어 문장을 한국인들이 더 쉽게 이해할 수 있도록 현지화해 주세요.
                      """),
        SystemMessage(content="문장이 수정된 이유를 반드시 한국어로 함께 설명하세요."),
        HumanMessage(content=state["text"])  # ✅ State 의 'text' 키 사용
    ]
    localized_text = localization_llm(prompt).content
    return {"text": localized_text}  # ✅ 딕셔너리 형태로 반환
```

4. Node 추가

```python
# Node 추가
graph.add_node("initial_translation", initial_translation)
graph.add_node("localization", localization)
```

```python
from langchain.chat_models import ChatOpenAI
import os

initial_translation_llm = ChatOpenAI(
    model_name="deepseek/deepseek-chat:free",
    temperature=0,
    openai_api_base="https://openrouter.ai/api/v1",
    openai_api_key=os.environ["OPENROUTER_API_KEY"]
)

localization_llm = ChatOpenAI(
    model_name="deepseek/deepseek-chat:free",
    temperature=0.7,
    openai_api_base="https://openrouter.ai/api/v1",
    openai_api_key=os.environ["OPENROUTER_API_KEY"]
)
```

LLM 정의

LangGraph

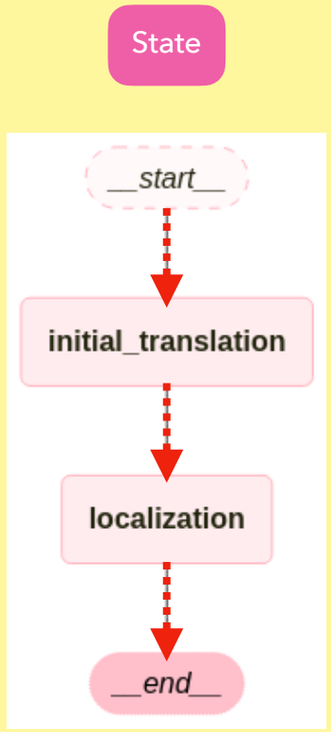
구현 예시 : Edge 연결 / 그래프 컴파일

5. Edge 연결

```python
# Edge 연결
graph.set_entry_point("initial_translation")
graph.add_edge("initial_translation", "localization")
graph.add_edge("localization", END)  # 답변 -> 종료
```

6. 그래프 컴파일

```python
# 그래프 컴파일
graph = graph.compile()
```



LangGraph

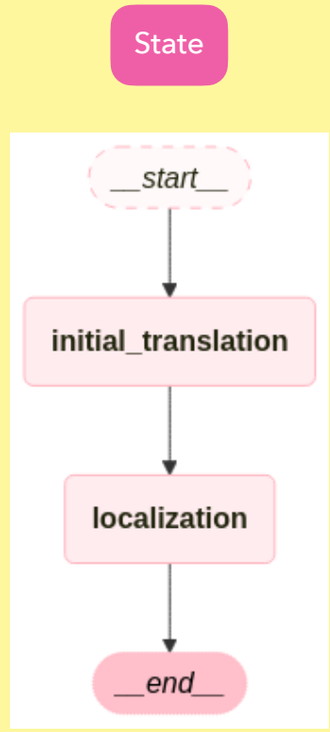
구현 예시 : 실행 & 결과

7. 그래프 실행

```python
# 입력 텍스트
input_text = "What were you guys smokin' when you came up with that?"

# 초기 상태 설정
state = {"text": input_text}

# 그래프 실행 및 결과 출력
for result in graph.stream(state):
    print(f"🔄 단계: {list(result.keys())[0]}")  # 현재 실행 중인 노드 이름
    print(f"📝 결과: {result[list(result.keys())[0]]['text']}\n")  # 해당 노드의 결과값
```



실행 결과

```
🔄 단계: initial_translation
📝 결과: 그걸 생각해낼 때 뭘 피우고 있었어?

🔄 단계: localization
📝 결과: "그걸 생각해냈을 때 무슨 생각을 하고 있었어?"

수정 이유:
1. "뭘 피우고 있었어?"는 한국어에서 '무엇을 흡연하고 있었는가'라는 의미로 해석될 수 있어 원래 의도와 다를 수 있습니다.
2. 대신 '무슨 생각을 하고 있었어?'라고 수정하면, 아이디어를 떠올릴 때 어떤 생각을 하고 있었는지 물어보는 자연스러운 표현이 됩니다.
```

이상하네 🤔
뭐가 문제지

- 번역가는 의역할 때 원문을 충분히 참고하고 분석한다

번역가가 창조적으로 의역할 때, 원문을 참고하는 것이 매우 중요합니다. 원문을 그대로 번역하는 것만으로는 때로는 그 의미나 뉘앙스를 잘 전달하기 어려운 경우가 많죠. 그래서 창조적인 의역은 원문의 핵심적인 메시지나 감정, 분위기를 유지하면서도 대상 언어나 문화에 맞게 적절히 변형하는 과정입니다.

하지만 그 과정에서도 원문을 충분히 분석하고, 어떤 부분을 변형할지, 어떤 부분은 그대로 유지할지를 신중하게 결정해야 합니다. 원문을 완전히 무시하거나 지나치게 창의적인 해석을 한다면 원래 의도와 멀어질 수 있기 때문에, 항상 원문을 기반으로 하되, 번역가는 대상 언어의 특성이나 문화를 반영하여 더 자연스럽고 효과적으로 전달할 수 있도록 해야 합니다.

ChatGPT

LangGraph

번역 과정에서 원문을 참고하도록 수정

- State key 수정

```python
class State(TypedDict):
    text: str
```

```python
class State(TypedDict):
    origin_text: str
    output_text: None|str
```

- Node 입/출력 key 수정

```python
# 첫 번째 노드: 초벌 번역 (영어 → 한국어)
def initial_translation(state: State):
    prompt = [
        SystemMessage(content="당신은 콘텐츠 전문 번역가입니다. 사용자가 제공하는 문장을 한국어로 번역하세요."),
        HumanMessage(content=state["text"])  # ✅ State 의 'text' 키 사용
    ]
    translated_text = initial_translation_llm(prompt).content
    return {"text": translated_text}  # ✅ 딕셔너리 형태로 반환

# 두 번째 노드: 현지화 (자연스러운 한국어로 수정)
def localization(state: State):
    prompt = [
        SystemMessage(content="""
                      당신은 한국어 콘텐츠 현지화 전문가입니다.
                      초벌 번역된 한국어 문장을 한국인들이 더 쉽게 이해할 수 있도록 현지화해 주세요.
                      """),
        SystemMessage(content="문장이 수정된 이유를 반드시 한국어로 함께 설명하세요."),
        HumanMessage(content=state["text"])  # ✅ State 의 'text' 키 사용
    ]
    localized_text = localization_llm(prompt).content
    return {"text": localized_text}  # ✅ 딕셔너리 형태로 리턴
```

```python
# 첫 번째 노드: 초벌 번역 (영어 → 한국어)
def initial_translation(state: State):
    prompt = [
        SystemMessage(content="당신은 콘텐츠 전문 번역가입니다. 사용자가 제공하는 문장을 한국어로 번역하세요."),
        HumanMessage(content=state["origin_text"])  # ✅ State 의 'text' 키 사용
    ]
    translated_text = initial_translation_llm(prompt).content
    return {"origin_text": state["origin_text"], "output_text": translated_text}  # ✅ 딕셔너리 형태로 리턴

# 두 번째 노드: 현지화 (자연스러운 한국어로 수정)
def localization(state: State):
    prompt = [
        SystemMessage(content="""
                      당신은 한국어 콘텐츠 현지화 전문가입니다.
                      초벌 번역된 한국어 문장을 한국인들이 더 쉽게 이해할 수 있도록 현지화해 주세요.
                      """),
        SystemMessage(content="문장이 수정된 이유를 반드시 한국어로 함께 설명하세요."),
        HumanMessage(content=f"원문: {state['origin_text']}\n초벌 번역: {state['output_text']}")
    ]
    localized_text = localization_llm(prompt).content
    return {"origin_text": state["origin_text"], "output_text": localized_text}  # ✅ 딕셔너리 형태로 리턴
```

LangGraph

번역 과정에서 원문을 참고하도록 수정

- 그래프 실행 입/출력값 수정

```python
# 입력 텍스트
input_text = "What were you guys smokin' when you cane up with that?"

# 초기 상태 설정
state = {"text": input_text}

# 그래프 실행 및 결과 출력
for result in graph.stream(state):
    print(f"🔄 단계: {list(result.keys())[0]}")  # 현재 실행 중인 노드 이름
    print(f"📝 결과: {result[list(result.keys())[0]]['text']}\n")  # 해당 노드의 결과값
```

```python
# 입력 텍스트
input_text = "What were you guys smokin' when you came up with that?"

# 초기 상태 설정
state = {"origin_text": input_text}

# 그래프 실행 및 결과 출력
for result in graph.stream(state):
    print(f"🔄 단계: {list(result.keys())[0]}")  # 현재 실행 중인 노드 이름
    print(f"📝 결과: {result[list(result.keys())[0]]['output_text']}\n")  # 해당 노드의 번역 결과
```

LangGraph
수정한 결과

단계: initial_translation
결과: 그걸 생각해낼 때 뭘 피우고 있었어?

단계: localization
결과: "그걸 생각해냈을 때 무슨 생각을 하고 있었어?"

원문을 참조하지 않고 현지화

수정 이유:
1. "뭘 피우고 있었어?"는 한국어에서 '무엇을 흡연하고 있었는가'라는 의미로 해석될 수 있어 원래 의도와 다를 수 있습니다.
2. 대신 '무슨 생각을 하고 있었어?'라고 수정하면, 아이디어를 떠올릴 때 어떤 생각을 하고 있었는지 물어보는 자연스러운 표현이 됩니다.

단계: initial_translation
결과: 그걸 생각해낼 때 뭘 피우고 있었어?

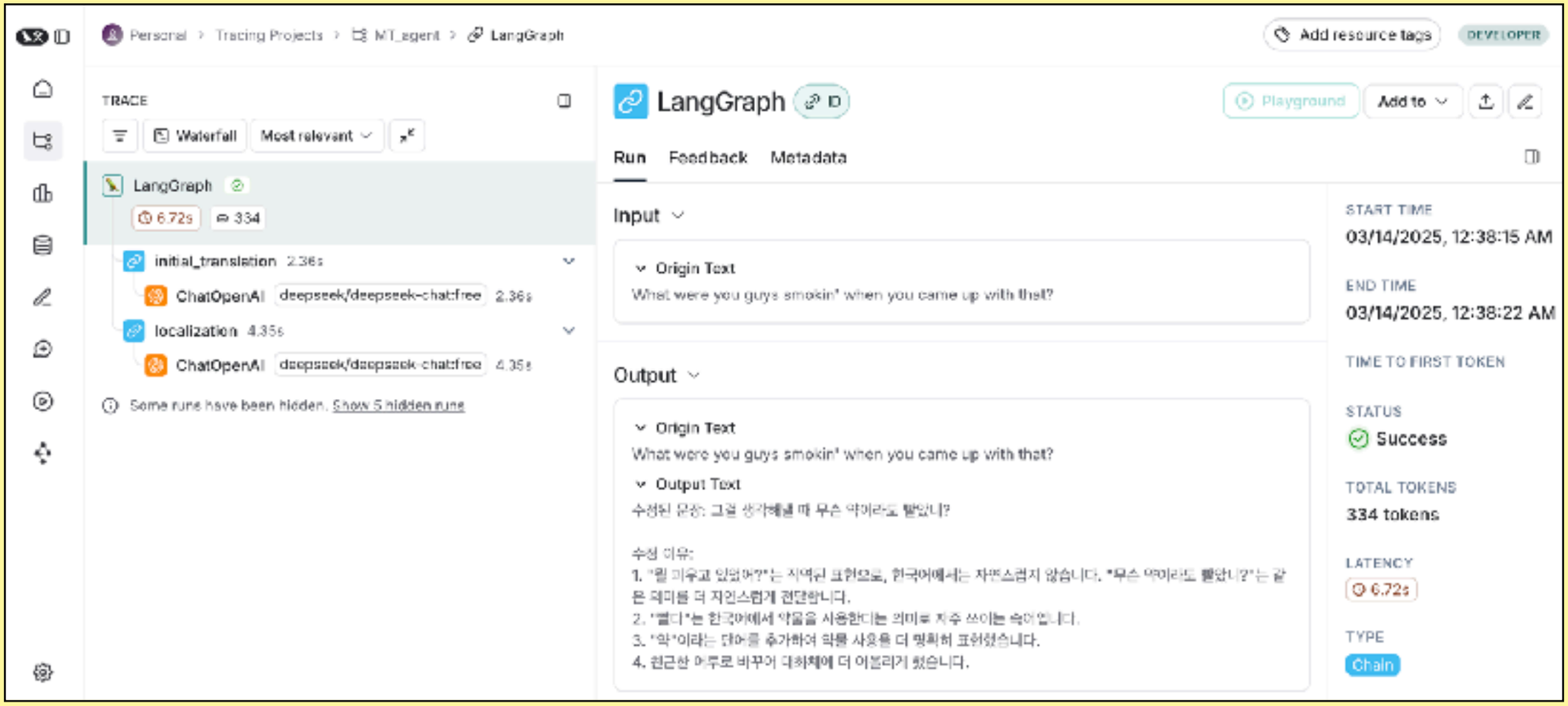
단계: localization
결과: 수정된 문장: 그걸 생각해낼 때 무슨 약이라도 했었니?

원문을 참조하여 현지화

수정 이유:
1. "뭘 피우고 있었어?"는 직역된 표현으로, 한국어에서는 자연스럽지 않습니다. "무슨 약이라도 했었니?"는 같은 의미를 더 자연스럽게 전달합니다.
2. "빌디"는 한국어에서 약물을 사용한다는 의미로 자주 쓰이는 속어입니다.
3. "약"이라는 단어를 추가하여 약물 사용을 더 명확히 표현했습니다.
4. 친근한 어투로 바꾸어 대화체에 더 어울리게 했습니다.

Tip : LangSmith

LLM 모니터링 도구



Agent 가 최선일까?

LLM 1개만으로 초월번역을 해보자

```python
from langchain.chat_models import ChatOpenAI
from langchain.schema import SystemMessage, HumanMessage
import os


text = "What were you guys smokin' when you came up with that?"

client = ChatOpenAI(
    model_name="deepseek/deepseek-chat:free",
    temperature=0.7,
    openai_api_base="https://openrouter.ai/api/v1",
    openai_api_key=os.environ["OPENROUTER_API_KEY"]
)

prompt = [
    SystemMessage(content="""
            당신은 콘텐츠 전문 번역가이자 현지화 전문가입니다. 사용자가 제공하는 문장을 한국어로 번역하세요.
            번역 과정에서 한국인들이 더 쉽게 이해할 수 있도록 현지화해 주세요.
            문장이 수정된 이유를 반드시 한국어로 함께 설명하세요.
            """),
    HumanMessage(content=f"{text}")
]


print(client.invoke(prompt).content)
```

LLM 추론 결과

"그걸 생각해 낼 때 뭘 빨고 있었어?"

번역 이유:

1. "What were you smokin'"은 비속어 표현으로,
무언가 비현실적이거나 이상한 아이디어를 냈을 때 사용하는 표현입니다.
한국어로는 "뭘 빨고 있었어"라고 번역하여 비슷한 느낌을 전달했습니다.

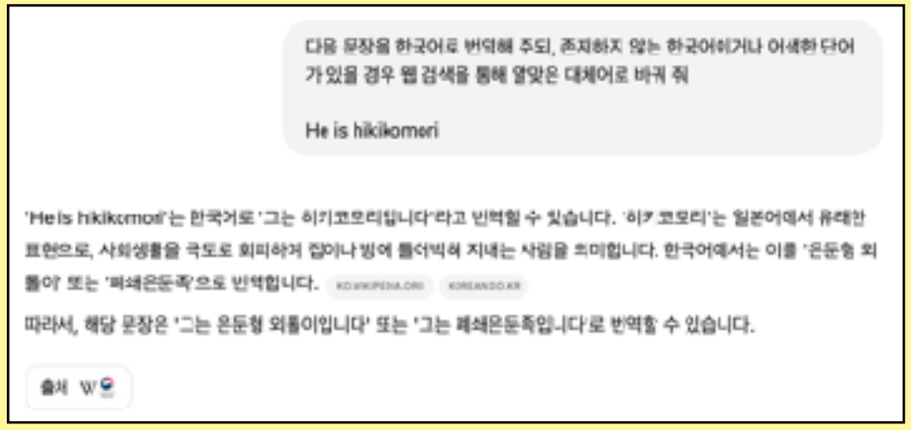
2. "you guys"는 "너희들"로 번역할 수 있지만, 한국어에서는 생략해도 자연스러운 경우가 많아 생략했습니다.

3. 전체적으로 비속어 표현을 살리면서도 한국인들이 쉽게 이해할 수 있도록 현지화했습니다.

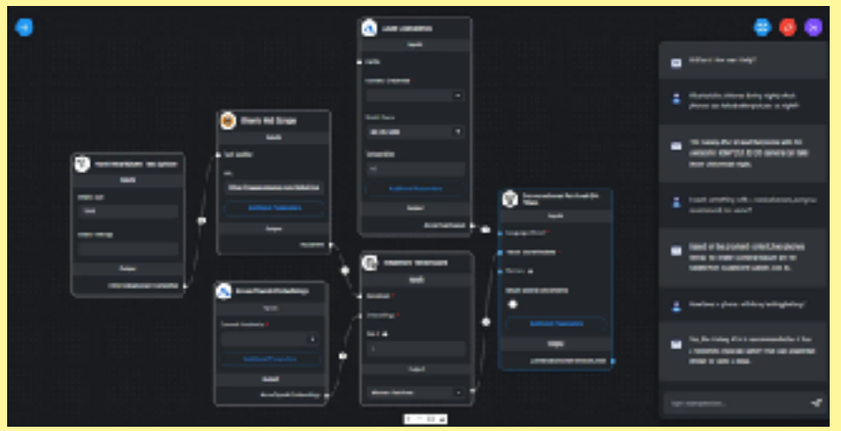
초월번역기 개발 회고

무슨 마약했길래 이런 생각을 했을까

- 단순하고 순차적인 프로세스를 처리할 거라면 Agent 대신 LangChain 을 써야 했다
    - Agent 는 복잡한 작업을 처리할 때, AI가 자율적으로 작업을 결정해야 할 때 유용함
- 차라리 신조어 번역기를 만들었어야 했다
    - 웹 검색 tool 을 사용해서 새로운 단어도 번역할 수 있도록.



- 코딩부터 하는 대신 No-Code 툴로 먼저 빠르게 개발하고 효과를 검증했어야 했다
    - Make, Zapier, Flowise…



AI Agent 개발 회고

AI는 개발자를 대체할 수 없다!

- 전문적인 Agent 를 구축하기 위해서는 도메인에 대한 깊은 이해 필요
  - 개발자가 번역 산업에 대한 이해도가 낮았다면 TransAgent 를 개발할 수 없었을 것.
  - 이는 AI가 대체할 수 없음 (Agent 를 개발하는 Agent 가 나온다면 과연…?)

- 복잡한 작업을 수행하는 Agent 를 구현하는 능력 == 설계 능력
  - 각각의 Agent 가 상호작용하려면?
    - 단순 파이프라인? 조건문? 작업 분배? …
  - 입/출력 데이터는 무엇이 필요한가?

우린 괜찮을 거야
마무리를 어떻게 하지

- 새로운 기술이 엄청나게 빠르게 쏟아져 나온다 ➡️ 지속적으로 학습하고 변화에 적응하자

- AI 는 결국 문제를 해결하기 위한 도구이다 ➡️ 문제를 정확히 이해하고, 해결하기 위한 개발 및 설계에 집중하자

- AI 프로젝트는 보통 다양한 전문가들이 함께 작업한다 ➡️ 협업과 커뮤니케이션 능력을 기르자

- AI 프로젝트 개발 및 유지에는 비용과 자원이 많이 든다 ➡️ 효율화 및 최적화에 힘쓰자

들어주셔서 감사합니다