Report: Comparison of OpenAI API vs Microsoft Azure SDK for translation

June 30, 2025

Objective

This report compares OpenAI's API (primarily GPT-3 and GPT-4) with Microsoft Azure SDK's Translator API for text translation tasks. The comparison focuses on translation quality, cost, and response time (latency) to help identify the best solution based on specific use cases.

1. **Overview of Services:**

Criteria	OpenAI API	Microsoft Azure SDK
Service type	Language model for flexible translation	Dedicated translation service
Supported languages / dialects (Portugal – BR, Spanish – MX, CO, CL & AR)	All; need prompt adjustment	All; need pre-classification. Only formal tone
Target use case	Wide	Formal
Handling of regional dialects	Good at handling sub-regional slang, idioms and conversional tone	Neutralizes sub-regional dialect, focuses on accurate translations
Translation focus	Adaption to tone, context & nuance	Precision in formal & technical translations

2. **Cost Comparison:**

Criteria	OpenAI API	Azure SDK
Pricing model	Token-based	Character-based
Cost per 1,000 words	??	??
Cost per 10,000 words	??	??
Cost advantage	Could be more expensive for larger text	More cost-effective for bulk translation due to character based pricing. *(Need confirmation)*

3. **Time lag (latency) Comparison:**

Criteria	OpenAI API	Microsoft Azure SDK
Typical latency	0.2 ~ 3 seconds (token size dependent)	0.1 ~ 1 seconds (faster for bulk, connection speed dependent)
Factors affecting latency	Longer prompts Model size (GPT-4,GPT-4o, etc) Server load	Character count, Server region, API Load
Speed	Good for short to medium text, slow for large inputs	Faster for large-scale. Optimized for large-volume

4. **Strengths & weakness:**

Criteria	OpenAI API	Microsoft Azure SDK
Strengths	Handles informal, creative & conversational tone. Good for transliterations with cultural awareness Adaptable & scalable quickly to multiple dialects.	Ideal for formal translations. High precision for technical & business translations Bulk translation tasks are optimized
Weakness	More expensive for large text volumes. Slower responses.	Sub-regional dialects were neutralized. Informal tone is removed. Less flexible for creative content. (Don't know how it will react to insurance claims.)

5. **Dialects & Slang handling:**
   The regional dialect comparison was compared with the translated data.

Region	OpenAI API	Azure SDK
Mexican	Captures colloquial tone and idioms like "echar la flojera" (to chill) and "¿te animas?" (feel like it?), translating them naturally into informal English. Preserves warmth and informality.	Tends to literalize idioms ("be lazy" for "echar la flojera"), making the translation technically accurate but less culturally natural.
Argentinian	Understands distinctive Rioplatense features like "che" (attention-grabbing interjection), "pibes" (kids/friends), and "te prendes?" (wanna join?). Outputs relaxed, local-sounding English.	Translates with neutral vocabulary, losing cultural markers. For example, "pibes" becomes "friends" or "guys", and "che" is omitted entirely.
Brazillian*	Interprets casual syntax and omissions (e.g., dropping articles, relaxed phrasing) as part of a spoken register, producing friendly, idiomatic English.	Translates more formally; converts informal structures to grammatical English, which can feel stiff or over-polished in conversational settings.
Colombian	Preserves the politeness and structure of Bogotian Spanish while adapting naturally. Handles phrases like "¿quieres acompañarme?" as "Want to come with me?"—still polite, but fluid.	Retains the grammatical formality—e.g., "Do you want to accompany me?"—which is correct but less typical in natural spoken English.
Chilean	Translates Chilean idioms like "¿te tinca?" (sound good to you?) and "cabros" (kids/friends) with creative equivalence. Keeps the playful, youthful tone	Provides a neutral, direct translation—e.g., "¿te tinca?" → "Does it sound good to you?"—which is accurate but flattens the local dialect.

Spanish dialect direct comparison with the values, with some examples from the dataset made.
General translation. "Hey, I'm going to the cinema with my friends over this weekend. Do you want to come with us?"

Region	Sample sentence	Dialect features	OpenAI Output	AzureSDK
Mexico	*"Voy a echar la flojera este finde, ¿te animas a ver unas pelis?"*	"Echar la flojera" = chill, "pelis" = movies (slang)	"I'm going to chill this weekend, want to watch some movies?"	"I'm going to be lazy this weekend; do you want to watch some movies?"
Argentina	*"Che, este finde voy al cine con unos pibes. ¿Te prendes?"*	"Che" = interjection, "pibes" = kids/friends, "prenderse" = join	"Hey, this weekend I'm going to the movies with some friends. Want to join?"	"I am going to the movies with some friends this weekend. Would you like to come?"
Brazil*	*"Voi ao cinema com uns amigo, esse final de semana, quer vir?"*	Informal Brazillian Portuguese	"I'm going to the movies with some friends this weekend. Want to come?"	"I am going to the movies with some friends this weekend. Would you like to come?"
Colombia	*"Voy a ir al cine con unos amigos este fin de semana, ¿quieres acompañarme?"*	Polie, standard Bogotian Spanish	"I'm going to the movies with some friends this weekend, want to come with me?"	"I am going to the movies with some friends this weekend. Do you want to accompany me?"
Chile	*"Voy a ir al cine con unos cabros este fin de semana, ¿te tinca?"*	"Cabros" = kids/friends, "te tinca" = does it sound good?	"I'm going to the movies with some friends this weekend. Sound good to you?"	"I am going to the movies with some friends this weekend. Does it sound good to you?"

Scoring summary

- Regional slang accuracy: OpenAI API
- Formality Handling: Azure SDK
- Context preservation: OpenAI API
- Tone preservation: OpenAI API
- Literal accuracy: Azure SDK
- Creative Adaptation: OpenAI API

6. **Conclusions**

Concluding, against the task of translating different Spanish dialects—like those from Mexico, Argentina, Colombia, and Chile—OpenAI API performed better capturing the local tone, slang, and natural flow of conversation. It translates phrases in a way that sounds more like how people do actually speak, which makes it great for things like apps, chats, or anything meant for real users. Microsoft Azure's translator SDK, on the other hand, gives us correct and clear translations, but they're usually more formal and can feel a bit flat or generic. It's better suited for business documents or official materials. So, if you need accurate but lively, natural-sounding English, OpenAI is often the better choice. But if for the need of speed, structure, and formality, Azure does the job well.

I also suggest using SDK in English to Spanish translations as the conversation to the customer should be Formal & respectful, however the customer base could be a bit shaky so

OpenAI API could serve use better as well. Purely monetary wise, I recommend using a hybrid translation workflow, however this will depend on the net monetization of the OpenAI API and Microsoft Azure cost. This could strike a good balance between accuracy, speed & cost.

At the initial intake stage, where customers often submit informal descriptions—possibly containing regional slang or voice-to-text input—OpenAI's API (such as GPT-4 or GPT-4o) is ideal, as it handles dialects and conversational tone with high sensitivity. Once claims progress to document-heavy stages like ID verification, police reports, or structured forms, Microsoft Azure's Translation SDK becomes more effective due to its speed and accuracy in processing formal, repetitive text at scale. For internal adjuster notes or summaries, either tool may be used depending on tone—OpenAI if nuance and readability matter, Azure if formality is key. Finally, for compliance-ready output and official filings, Azure should be the default due to its consistent, business-suitable translations. This mix-and-match approach ensures the best translation quality where it matters, while optimizing cost and throughput.