Twilio Docs

On this page

Best practices

Streaming text tokens to ConversationRelay

If your application uses an LLM API then you likely have the option of receiving text tokens streamed to you in chunks from the LLM provider. Most applications should stream these text tokens to ConversationRelay as soon they become available instead of waiting for the LLM to generate a complete response. This will allow ConversationRelay to begin speaking the response to your caller sooner.

Alternatively, you can wait for the LLM to finish generating the entire response before sending it to ConversationRelay. This approach can introduce significant latency but it might offer a smoother and more consistent pace of speaking depending on your application and LLM. Test both approaches to see which works best for your use case.

Testing different speech-to-text providers

It's important to try different speech-to-text (STT) providers and models to find the best fit for your application's needs. Google and Deepgram each offer unique strengths for different scenarios, such as clean versus noisy audio environments. You can configure your STT provider and model using the `ovider` and `speechMode` attributes in the `ay>` noun in TwiML.

Skip to content Skip to navigation Skip to topbar

Normalizing text for TTS

formats. Consider the following guidelines:

- **Numbers and units**: Write numbers as words for improved pronunciation. For example, write "twenty dollars and fifty cents" instead of "$20.50".
- **Dates**: Spell out dates completely (for example, "March twenty-eighth, two thousand twenty-five" instead of "03/28/2025").
- **Email addresses**: Replace or spell out special characters. For instance, write "user at example dot com" rather than "user@example.com" since the "@" sign may be mispronounced.
- **Names**: Use consistent formatting for names throughout the call to avoid variations (for example, always use "Anna" rather than alternating between "A-nna" and "Ah-nna").
- **Abbreviations**: Spell out abbreviations that should be pronounced fully (for example, "Doctor" instead of "Dr.").
- **Special characters**: Replace special characters with their spoken equivalents (for example, "percent" instead of "%").
- **Punctuation**: Use appropriate punctuation to create natural pauses and intonation.
- **Acronyms and initials**: Insert spaces between letters if they must be spelled out (for example, "H T T P" for letter-by-letter pronunciation).

If you use the ElevenLabs TTS provider then you can also use the `elevenlabsTextNormalization` TwiML attribute to help with text normalization. Setting this attribute to `on` will tell ElevenLabs to take an extra step to normalize text for you. Set the attribute to `off` for more control over the text normalization process and lower latency.

For detailed text normalization guidelines, refer to ElevenLabs' text normalization best practices .

Monitor for error messages from the ConversationRelay WebSocket connection. These messages can indicate a problem with the message that you sent us or an issue with the connection. Implement appropriate error handling in your application to manage these situations gracefully.

Handling language switching

Use the `language` message to switch languages dynamically during a session. When you switch languages during a session via a WebSocket message, ConversationRelay uses the pre-set voice associated with the new language that you configured in the initial TwiML setup. If you didn't configure a specific voice for that language, ConversationRelay will use its default voice for the selected language.

- **Initial Language Configuration**: To maintain a consistent voice experience across languages, define the desired voice for each supported language explicitly in the TwiML configuration. This setup ensures that when you change the language via a WebSocket message during the session, the specified voice is used. This prevents the system from defaulting to its own voice.
- **Voice on Language Change**: If you issue a language change and haven't set a specific voice for that language in TwiML, the system will use its default voice for that language. For example, switching from English (en-GB) with a male voice to Spanish (es-ES) may result in a female voice if the default Spanish voice is female.
- **No Voice Updates Mid-Session**: Once the session starts, you can't modify voice and language configurations set in TwiML ket messages.

This setup ensures consistent voice behavior for each language by configuring it in TwiML before the call begins.

when sending text tokens to ConversationRelay

In streaming mode

Send each text token incrementally with `"last": false`. When the LLM indicates that the response is complete (for example, when `response.finish_reason()` equals `"stop"`), mark that final token with `"last": true`. For messages with complex punctuation, consider breaking the response into smaller chunks.

Example:

```
{ "type": "text", "token": "Hello", "last": fa
{ "type": "text", "token": " world", "last": f
{ "type": "text", "token": "!", "last": true }
```

In non-streaming mode

When the entire response is generated as a single complete sentence, mark the token with `"last": true`.

Example:

```
{ "type": "text", "token": "Hello world!"
```

Implementing agent observability with Conversational Intelligence

enable this feature, you need to set the `intelligenceService` attribute in the `<ConversationRelay>` noun as documented below.

For more details on getting started with the ConversationRelay and Conversational Intelligence integration, please see this guide.