

# How to solve common chatbot issues

## 1 Introduction

Chatbots are everywhere and there is almost no company that is not considering to develop its own to better communicate with its customers.

However, while exploring different options you may have realized that most of the bot development platforms have forgotten about the importance of Natural Language Processing, a key component to create a human-like chatbot able to understand customers and answer them properly.

For a better understanding of the issues we did some testing of bots and all the major chatbot development platforms like Wit.ai, Api.ai and LUIS, and as we expected we found some issues related to language that haven't been solved for years and can be easily resolved by using a linguistic component.

## 2 Common issues detected

### 2.1 Negation

Many bots don't understand negation in a phrase because they have been built based on a keyword approach. That makes it difficult for users to ask for something as simple as "I want a barbeque pizza with no pork". Let's see some examples:

- "I want a barbeque pizza with **no** pork" (only negates pork).
- "We **don't** want any drinks" (negates the whole event).
- "I'm **not** sure... I'll take a beer (It doesn't negate the main event)"

```
{"action":"complex.order","parameters":{"topping.add":["null"],"topping.neg":[],"type":["barbeque"]},"resolvedQuery":"I want a barbaque pizza with no pork"}
```

Example of how one of the major chatbot development platforms doesn't handle negation accurately.

## 2.2 Coordination

It is one of the most used elements in how humans talk, and after our research we found out that most relevant platforms do not support a request where elements are joined by a coordinator. Our linguistic knowledge makes us capable to solve this issue.

- “[I want a Hawaiian pizza] and [my wife will have a Margherita]” (**two main events**).
- “I’ll have a Hawaiian [with [extra cheese] and [onion]]” (**two changes in ingredients**).
- “I’ll take [[a Hawaiian [with [extra cheese] and [onion]]] and [a Margherita]]” (**two pizzas, the first one with two ingredients**)

```
{
  "action": "complex.order",
  "parameters": {
    "topping.add": ["my wife"],
    "topping.neg": [],
    "type": ["hawaiian"],
    "resolvedQuery": "I want a Hawaiian pizza, and my wife will have a Margherita"
  }
}
```

Example of how one of the major chatbot development platform doesn't handle coordination accurately.

## 2.3 Double intent

Most chatbot frameworks are based around the concept of intent and entity detection, which involves identifying both the intent of an utterance and the entities relevant to that intent. For example, for the sentence “I want a pepperoni pizza”, most chatbot frameworks – after being properly configured and trained – would detect “order food” as the intent, and “pepperoni pizza” as the “food type” entity.

For simple utterances such as the previous example, most chatbot frameworks work correctly. But when users ask for assistance with more complex requests, existing solutions are often not able to cope.

For example, consider the utterance “I want a pepperoni pizza and a soda”, which has two entities “pepperoni pizza” and “soda” which could be the object of the intent. Most frameworks only support a single entity for each intent, so they cannot easily handle natural requests with two intents such as this one. **The problem is known as “double intent”.**

This is usually a design limitation, because intent detection is typically handled as a text classification problem, and text classification models are designed to output a single class for a given text. For example, for the sentence “I want to order a pizza and rent a movie”, there are two separate intents: “order food” and “rent movie”.

At Bitext we have developed a solution to this problem. Using our Deep Linguistic Analysis Platform we rewrite the input request into distinct sentences with one intent each. This allows existing chatbot frameworks to deal with this type of user requests without having to redesign their architectures. Instead, our system takes “I want to order a pizza and rent a movie” and outputs “I want to order a pizza” and “I want to rent a movie”, which can be processed sequentially by existing frameworks.

Our Platform also allows existing frameworks to deal with other complex linguistic phenomena like negation (“I want a deluxe pizza without mushrooms”) or conditional structures (“If it is included in the offer, add a beer”) without having to add extensive intent detection rules and hundreds of examples for training. Our Platform provides these services mainly through its syntactic component or parser, available in more than 20 languages.

```
{"action":"complex.order";"parameters":{"topping.add":[],"topping.neg":[],"type":["order"]},"resolvedQuery":"I want to order a pizza and rent a movie"}
```

Example of how one of the major chatbot development platform doesn't handle double intent accurately.

### 3 Conclusion

Our linguistic platform is able to solve language complexity and increase by 90% the answer accuracy in any major chatbot development platform, shortening training process up to 60% and improving user satisfaction .