

Bitext Natural Language Generation (NLG) for any chatbot platform

Bitext NLG is our solution to speed the training of any chatbot, automatically generating loads of different training sentences with the same meaning as the original one, so allowing any chatbot developer to write manually very few sentences per intent and obtain automatically thousands of different sentences. The generation of Artificial Training Data (also called Variant Generation) is one of the most successful ways to speed up the setup of a chatbot with a minimum effort.

One of the most interesting features of Bitext NLG is that it not only generates the different sentences: it also tags their intent and slots/entities and delivers them in the format that the most important chatbot platforms are expecting. This way you can immediately use the dataset generated by Bitext NLG to train your chatbot.

As Bitext NLG supports many different platforms' output, you can generate your training data for all different platforms at once, thus giving you the power to quickly switch from one platform to another one, or even maintain several chatbots at a time, all of them with the same training data, and see what behaves better for you.

In this document we'll give some examples of how Bitext NLG can generate the same sentence with the specific tagging and format of all the main chatbot platforms.

1. **Rasa**
2. **MS Luis**
3. **Dialogflow**
4. **Lex**
5. **Wit.ai**
6. **Watson**

Sample sentence: **"please, can you turn on the kitchen lights?"**

1. Rasa

```
{
  "text": "please, can you turn on the kitchen lights?",
  "entities": [
    {
      "entity": "Action",
      "start": 16,
      "end": 23,
      "value": "turn on"
    },
    {
      "entity": "Place",
      "start": 28,
      "end": 35,
      "value": "kitchen"
    },
    {
      "entity": "Object",
      "start": 36,
      "end": 42,
      "value": "lights"
    }
  ],
  "intent": "turn_on"
}
```

2. MS Luis

```
{
  "text": "please, can you turn on the kitchen lights?",
  "intentName": "turn_on",
  "entityLabels": [
    {
      "startCharIndex": 16,
      "endCharIndex": 22,
      "entityName": "Action"
    },
    {
      "startCharIndex": 28,
      "endCharIndex": 34,
      "entityName": "Place"
    },
    {
      "startCharIndex": 36,
      "endCharIndex": 41,
      "entityName": "Object"
    }
  ]
}
```

3. Dialogflow

```
[
  {
    "count":0,
    "data":[
      {
        "text":"please, can you ",
        "userDefined":false
      },
      {
        "text":"turn on",
        "alias":"Action",
        "meta":"@Action",
        "userDefined":true
      },
      {
        "text":" the ",
        "userDefined":false
      },
      {
        "text":"kitchen",
        "alias":"Place",
        "meta":"@Place",
        "userDefined":true
      },
      {
        "text":" ",
        "userDefined":false
      },
      {
        "text":"lights",
        "alias":"Object",
        "meta":"@Object",
        "userDefined":true
      },
      {
        "text":"?",
        "userDefined":false
      }
    ]
  }
]
[
  {
    "entries":[
      {
        "value":"turn on"
      }
    ],
    "name":"Action"
  },
  {
    "entries":[
      {
        "value":"kitchen"
      }
    ],
  },
]
```

```
    "name": "Place"  
  },  
  {  
    "entries": [  
      {  
        "value": "lights"  
      }  
    ],  
    "name": "Object"  
  }  
]
```

4. Lex

```
{
  "metadata": {
    "schemaVersion": "1.0",
    "importType": "LEX",
    "importFormat": "JSON"
  },
  "resource": {
    "name": "BOT_NAME",
    "version": VERSION_NUMBER,
    "intents": [
      {
        "name": "turn_on",
        "version": "1",
        "fulfillmentActivity": {
          "type": "CodeHook"
        },
        "sampleUtterances": [
          "{action} {place} {object}"
        ],
        "slots": [
          {
            "name": "Action",
            "slotConstraint": "Optional",
            "slotType": "Action",
            "valueElicitationPrompt": {
              "messages": [
                {
                  "contentType": "PlainText",
                  "content": "Placeholder prompt"
                }
              ]
            },
            "maxAttempts": 2
          },
          {
            "priority": PRIORITY,
            "sampleUtterances": []
          }
        ],
        {
          "name": "Place",
          "slotConstraint": "Optional",
          "slotType": "Place",
          "valueElicitationPrompt": {
            "messages": [
              {
                "contentType": "PlainText",
                "content": "Placeholder prompt"
              }
            ]
          },
          "maxAttempts": 2
        },
        {
          "priority": PRIORITY,
          "sampleUtterances": []
        }
      ],
      {
        "name": "Object",
        "slotConstraint": "Optional",
        "slotType": "Object",
```

```

        "valueElicitationPrompt": {
          "messages": [
            {
              "contentType": "PlainText",
              "content": "Placeholder prompt"
            }
          ],
          "maxAttempts": 2
        },
        "priority": PRIORITY,
        "sampleUtterances": []
      }
    ]
  },
  "slotTypes": [
    {
      "name": "Action",
      "version": VERSION_NUMBER,
      "enumerationValues": [
        {
          "value": "turn on"
        }
      ],
      "valueSelectionStrategy": "TOP_RESOLUTION"
    },
    {
      "name": "Place",
      "version": VERSION_NUMBER,
      "enumerationValues": [
        {
          "value": "kitchen"
        }
      ],
      "valueSelectionStrategy": "TOP_RESOLUTION"
    },
    {
      "name": "Object",
      "version": VERSION_NUMBER,
      "enumerationValues": [
        {
          "value": "lights"
        }
      ],
      "valueSelectionStrategy": "TOP_RESOLUTION"
    }
  ]
}

```

5. Watson

```
import json
import watson_developer_cloud

assistant = watson_developer_cloud.AssistantV1(
    username='USERNAME',
    password='PASSWORD',
    version='2018-02-16'
)

assistant.create_intent(
    workspace_id = 'WORKSPACE_ID',
    intent = 'turn_on',
    examples = [
        {'text': 'Please, can you turn on the kitchen lights?'}
    ]
)

assistant.create_entity(
    workspace_id = 'WORKSPACE_ID',
    entity = 'Action',
    values = [
        {'value': 'turn on'}
    ]
)

assistant.create_entity(
    workspace_id = 'WORKSPACE_ID',
    entity = 'Place',
    values = [
        {'value': 'kitchen'}
    ]
)

assistant.create_entity(
    workspace_id = 'WORKSPACE_ID',
    entity = 'Object',
    values = [
        {'value': 'lights'}
    ]
)
```

6. Wit.ai

```
{
  "text": "please, can you turn on the kitchen lights?",
  "entities": [
    {
      "entity": "intent",
      "value": "\"turn_on\""
    },
    {
      "entity": "Action",
      "value": "\"turn on\"",
      "start": 16,
      "end": 23
    },
    {
      "entity": "Place",
      "value": "\"kitchen\"",
      "start": 28,
      "end": 35
    },
    {
      "entity": "Object",
      "value": "\"lights\"",
      "start": 36,
      "end": 42
    }
  ]
}
```