

## Intro

- This intro code generates sample trees with the right structure. We generate nodes indicating whether it's a target (talking to a human, for example), with the text ('press 2 for x'), and a list of children for giving the tree structure.
- The code allows for customization of the branching factor, as well as the depth, leaving us with a wide family of trees
- GPT3 is used as the base model for generation. We might experiment with the new Qwen model that just got released.
- A drawback we have is that sometimes the targets aren't well identified, as seen in the examples generated.

## Code

```
import json
import os
import re
from typing import List, Optional
import openai
from pydantic import BaseModel, ValidationError

# Set your OpenAI API key
openai.api_key = 'SECRET'
if not openai.api_key:
    raise ValueError("OPENAI_API_KEY is not set. Please set it in your environment variables.")

# Define the MenuNode class
class MenuNode(BaseModel):
    number: int
    text: str
    is_target: bool
    children: List["MenuNode"] = []

class Config:
    arbitrary_types_allowed = True
    schema_extra = {
        "example": {
            "number": 1,
            "text": "For billing inquiries, press 1.",
            "is_target": False,
```

```

        "children": []
    }
}

# Update forward references for recursive MenuNode type
MenuNode.update_forward_refs()

# Utility function to extract JSON from text
def extract_json(text: str) -> Optional[str]:
    """
    Extracts the first JSON object or array from a string.

    Args:
        text (str): Input text.

    Returns:
        Optional[str]: JSON string if found, otherwise None.
    """
    try:
        match = re.search(r'(\{.*\}|\[.*\])', text, re.DOTALL)
        if match:
            json_str = match.group(0)
            # Validate the JSON
            json.loads(json_str)
            return json_str
        return None
    except json.JSONDecodeError:
        return None

# Function to generate children nodes using GPT
def generate_children(path: List[str], branching_factor: int, target_number: int) -> List[MenuNode]:
    """
    Generates structured child nodes for a given parent node using GPT.

    Args:
        path (List[str]): Path to the current node.
        branching_factor (int): Number of child nodes to generate.
        target_number (int): Number of 'is_target' nodes among the children.

    Returns:
        List[MenuNode]: Generated child nodes.
    """

```

```
"""
```

```
try:
```

```
    system_prompt = f"""
```

```
        You are an assistant generating a structured call center menu tree.
```

```
    ### Task:
```

```
    Generate exactly {branching_factor} child menu nodes for the current menu node.
```

```
    ### Requirements for Each Child Node:
```

- `number`: An integer corresponding to the option number (e.g., 1, 2, 3).
- `text`: A string describing the menu option (e.g., "For billing inquiries, press 1.").
- `is\_target`: A boolean indicating if this option leads directly to an agent (`true`) or not (`false`).
- `children`: An empty list (children will be generated recursively).

```
    ### Constraints:
```

- Exactly {target\_number} of the {branching\_factor} child nodes must have `is\_target` set to `true`.
- The `number` fields must be unique within the current set of children.
- The `text` field should clearly describe the action and include the press number.

```
    ### Current Path:
```

```
    { ' > '.join(path) if path else 'Root' }
```

```
    ### Output Format:
```

```
    Return a JSON array of the child nodes only. Do not include any additional text, explanations, or code blocks.
```

```
"""
```

```
response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[{"role": "system", "content": system_prompt}],
    temperature=0.7,
    max_tokens=500,
)
```

```
raw_content = response["choices"][0]["message"]["content"].strip()
```

```
# Extract JSON from the response
```

```
json_str = extract_json(raw_content)
```

```
if not json_str:
```

```
    raise ValueError("No valid JSON found in the response.")
```

```

children_data = json.loads(json_str)

# Validate and parse JSON into MenuNode objects
return [MenuNode(**child) for child in children_data]

except (json.JSONDecodeError, ValidationError, ValueError) as e:
    print(f"Error generating children for path {' > '.join(path) if path else 'Root'}: {e}")
    # Generate fallback child nodes in case of error
    return [
        MenuNode(
            number=i + 1,
            text=f"For option {i + 1}, press {i + 1}.",
            is_target=(i + 1 <= target_number),
            children=[]
        )
        for i in range(branching_factor)
    ]

# Function to recursively generate a menu tree
def generate_menu_tree(depth: int, branching_factor: int, target_number: int) -> MenuNode:
    """
    Recursively generates a menu tree structure.

    Args:
        depth (int): Depth of the tree.
        branching_factor (int): Number of children per node.
        target_number (int): Number of target nodes at each level.

    Returns:
        MenuNode: The root of the generated menu tree.
    """

    def build_tree(path: List[str], current_depth: int) -> MenuNode:
        is_root = current_depth == 0
        children = []
        if current_depth < depth:
            current_branching = 1 if is_root else branching_factor
            children = generate_children(
                path=path,
                branching_factor=current_branching,
                target_number=target_number
            )

```

```

    return MenuNode(
        number=1 if is_root else int(re.search(r'\d+', path[-1]).group()) if path else 1,
        text="Welcome to our service." if is_root else f"{path[-1]}",
        is_target=False,
        children=[
            build_tree(path + [child.text], current_depth + 1) for child in children
        ]
    )
return build_tree(path=[], current_depth=0)

# Function to export the menu tree to a JSON file
def export_menu_tree_to_json(tree: MenuNode, filename: str):
    """
    Exports the menu tree to a JSON file.

    Args:
        tree (MenuNode): The menu tree to export.
        filename (str): The output file name.
    """
    with open(filename, "w") as file:
        json.dump(tree.dict(), file, indent=4)
    print(f"Menu tree exported to {filename}")

# Main script
if __name__ == "__main__":
    # Define tree parameters
    tree_depth = 4
    branching_factor = 3
    target_number = 1 # Number of target nodes at each branching

    # Generate the menu tree
    menu_tree = generate_menu_tree(
        depth=tree_depth,
        branching_factor=branching_factor,
        target_number=target_number
    )

    # Export to JSON
    export_menu_tree_to_json(menu_tree, "menu_tree4.json")

```

## Appendex

### Example Trees generated

#### Tree 2

```
{
  "number": 1,
  "text": "Welcome to our service.",
  "is_target": false,
  "children": [
    {
      "number": 1,
      "text": "For option 1, press 1.",
      "is_target": false,
      "children": [
        {
          "number": 1,
          "text": "For option 1, press 1.",
          "is_target": false,
          "children": [
            {
              "number": 1,
              "text": "For option 1, press 1.",
              "is_target": false,
              "children": []
            },
            {
              "number": 2,
              "text": "For option 2, press 2.",
              "is_target": false,
              "children": []
            },
            {
              "number": 3,
              "text": "For option 3, press 3.",
              "is_target": false,
```

```
        "children": []
      }
    ]
  },
  {
    "number": 2,
    "text": "For option 2, press 2.",
    "is_target": false,
    "children": [
      {
        "number": 1,
        "text": "For option 1, press 1.",
        "is_target": false,
        "children": []
      },
      {
        "number": 2,
        "text": "For option 2, press 2.",
        "is_target": false,
        "children": []
      },
      {
        "number": 3,
        "text": "For option 3, press 3.",
        "is_target": false,
        "children": []
      }
    ]
  },
  {
    "number": 3,
    "text": "For option 3, press 3.",
    "is_target": false,
    "children": [
      {
        "number": 1,
        "text": "For option 1, press 1.",
```

```

        "is_target": false,
        "children": []
    },
    {
        "number": 2,
        "text": "For option 2, press 2.",
        "is_target": false,
        "children": []
    },
    {
        "number": 3,
        "text": "For option 3, press 3.",
        "is_target": false,
        "children": []
    }
]
}
]
}
]
}
]
}

```

### Tree 3

```

{
    "number": 1,
    "text": "Welcome to our service.",
    "is_target": false,
    "children": [
        {
            "number": 1,
            "text": "For customer service representative, press 1.",
            "is_target": false,
            "children": [
                {
                    "number": 1,
                    "text": "For technical support, press 1.",

```



```
"is_target": false,
"children": [
  {
    "number": 1,
    "text": "For software related issues, press 1.",
    "is_target": false,
    "children": []
  },
  {
    "number": 2,
    "text": "For hardware related issues, press 2.",
    "is_target": false,
    "children": []
  },
  {
    "number": 3,
    "text": "To speak directly with a technical support representative, press 3.",
    "is_target": false,
    "children": []
  }
]
},
{
  "number": 2,
  "text": "For billing inquiries, press 2.",
  "is_target": false,
  "children": [
    {
      "number": 1,
      "text": "For general billing questions, press 1.",
      "is_target": false,
      "children": []
    },
    {
      "number": 2,
      "text": "For specific invoice queries, press 2.",
      "is_target": false,
```

```
        "children": []
    },
    {
        "number": 3,
        "text": "For payment methods, press 3.",
        "is_target": false,
        "children": []
    }
]
},
{
    "number": 3,
    "text": "For questions about our products, press 3.",
    "is_target": false,
    "children": [
        {
            "number": 1,
            "text": "For information about product warranties, press 1.",
            "is_target": false,
            "children": []
        },
        {
            "number": 2,
            "text": "To speak with a product specialist, press 2.",
            "is_target": false,
            "children": []
        },
        {
            "number": 3,
            "text": "For information about product returns, press 3.",
            "is_target": false,
            "children": []
        }
    ]
}
]
```

```
]
}
```

## Tree 4

```
{
  "number": 1,
  "text": "Welcome to our service.",
  "is_target": false,
  "children": [
    {
      "number": 1,
      "text": "For immediate assistance from a customer service representative, press 1.",
      "is_target": false,
      "children": [
        {
          "number": 1,
          "text": "For billing inquiries, press 1.",
          "is_target": false,
          "children": [
            {
              "number": 1,
              "text": "For inquiries about your current bill amount, press 1.",
              "is_target": false,
              "children": [
                {
                  "number": 1,
                  "text": "To understand your bill breakdown, press 1.",
                  "is_target": false,
                  "children": []
                },
                {
                  "number": 2,
                  "text": "For payment options and details, press 2.",
                  "is_target": false,
                  "children": []
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

```
{
  "number": 3,
  "text": "To speak with an agent about your current bill amount, press 3.",
  "is_target": false,
  "children": []
}
],
{
  "number": 2,
  "text": "For inquiries about your previous bills, press 2.",
  "is_target": false,
  "children": [
    {
      "number": 1,
      "text": "For inquiries about the current bill, press 1.",
      "is_target": false,
      "children": []
    },
    {
      "number": 2,
      "text": "To dispute a charge on your previous bill, press 2.",
      "is_target": false,
      "children": []
    },
    {
      "number": 3,
      "text": "For all other bill-related questions, press 3.",
      "is_target": false,
      "children": []
    }
  ]
},
{
  "number": 3,
  "text": "To speak with a billing representative, press 3.",
  "is_target": false,
```

```
"children": [  
  {  
    "number": 1,  
    "text": "For questions about your bill, press 1.",  
    "is_target": false,  
    "children": []  
  },  
  {  
    "number": 2,  
    "text": "For payment arrangements, press 2.",  
    "is_target": false,  
    "children": []  
  },  
  {  
    "number": 3,  
    "text": "To speak with a billing representative, press 3.",  
    "is_target": false,  
    "children": []  
  }  
]  
}  
]  
,  
{  
  "number": 2,  
  "text": "For technical support, press 2.",  
  "is_target": false,  
  "children": [  
    {  
      "number": 1,  
      "text": "For software-related issues, press 1.",  
      "is_target": false,  
      "children": [  
        {  
          "number": 1,  
          "text": "For issues related to software installation, press 1.",  
          "is_target": false,  
          "children": []  
        }  
      ]  
    }  
  ]  
}
```

```
        "children": []
    },
    {
        "number": 2,
        "text": "For issues related to software updates, press 2.",
        "is_target": false,
        "children": []
    },
    {
        "number": 3,
        "text": "For issues related to software functionality, press 3.",
        "is_target": false,
        "children": []
    }
]
},
{
    "number": 2,
    "text": "For hardware-related issues, press 2.",
    "is_target": false,
    "children": [
        {
            "number": 1,
            "text": "For issues with computers, press 1.",
            "is_target": false,
            "children": []
        },
        {
            "number": 2,
            "text": "For issues with printers, press 2.",
            "is_target": false,
            "children": []
        },
        {
            "number": 3,
            "text": "If you're not sure what your hardware issue is, press 3.",
            "is_target": false,
```

```
        "children": []
      }
    ]
  },
  {
    "number": 3,
    "text": "To speak directly with a technical support representative, press 3.",
    "is_target": false,
    "children": [
      {
        "number": 1,
        "text": "For software related issues, press 1.",
        "is_target": false,
        "children": []
      },
      {
        "number": 2,
        "text": "For hardware related issues, press 2.",
        "is_target": false,
        "children": []
      },
      {
        "number": 3,
        "text": "To speak directly with a technical support representative immediately, press 3.",
        "is_target": false,
        "children": []
      }
    ]
  }
]
},
{
  "number": 3,
  "text": "To update your account information, press 3.",
  "is_target": false,
  "children": [
    {
```

```
"number": 1,
"text": "For changing account password, press 1.",
"is_target": false,
"children": [
  {
    "number": 1,
    "text": "To reset your password through email, press 1.",
    "is_target": false,
    "children": []
  },
  {
    "number": 2,
    "text": "To reset your password through text message, press 2.",
    "is_target": false,
    "children": []
  },
  {
    "number": 3,
    "text": "To speak to a customer service representative for password assistance, press 3.",
    "is_target": false,
    "children": []
  }
]
},
{
  "number": 2,
  "text": "To speak with a customer service representative, press 2.",
  "is_target": false,
  "children": [
    {
      "number": 1,
      "text": "For password reset, press 1.",
      "is_target": false,
      "children": []
    },
    {
      "number": 2,
```



```
        "text": "To update your billing information, press 2.",
        "is_target": false,
        "children": []
    },
    {
        "number": 3,
        "text": "To directly connect with a representative, press 3.",
        "is_target": false,
        "children": []
    }
]
},
{
    "number": 3,
    "text": "For updating email address linked to the account, press 3.",
    "is_target": false,
    "children": [
        {
            "number": 1,
            "text": "To confirm the update, press 1.",
            "is_target": false,
            "children": []
        },
        {
            "number": 2,
            "text": "To cancel the update and return to the previous menu, press 2.",
            "is_target": false,
            "children": []
        },
        {
            "number": 3,
            "text": "To speak with a representative about the email update, press 3.",
            "is_target": false,
            "children": []
        }
    ]
}
```

```
    ]  
  }  
  ]  
}  
]
```