

# Przewodnik importu i konfiguracji danych treningowych w Chatfuel

## Instrukcja techniczna

### Spis treści

1. [Przegląd procesu](#)
2. [Przygotowanie danych treningowych](#)
3. [Struktura bazy wiedzy w Chatfuel](#)
4. [Import danych treningowych](#)
5. [Konfiguracja intencji i encji](#)
6. [Testowanie i optymalizacja bazy wiedzy](#)
7. [Zarządzanie bazą wiedzy](#)
8. [Najlepsze praktyki](#)

## Przegląd procesu

Proces importu i konfiguracji danych treningowych w Chatfuel składa się z następujących etapów:

1. **Przygotowanie danych treningowych** - konwersja istniejącej bazy wiedzy NovaHouse do formatu akceptowanego przez Chatfuel
2. **Import danych** - załadowanie przygotowanych danych do platformy Chatfuel
3. **Konfiguracja intencji i encji** - definiowanie intencji użytkownika i encji do rozpoznawania
4. **Testowanie i optymalizacja** - weryfikacja poprawności rozpoznawania i odpowiedzi

## 5. Zarządzanie bazą wiedzy - procedury aktualizacji i rozszerzania bazy wiedzy

# Przygotowanie danych treningowych

## Analiza istniejącej bazy wiedzy

Wczesniej przygotowana baza wiedzy NovaHouse zawiera:

- FAQ (pytania i odpowiedzi)
- Opisy pakietów wykończeniowych
- Informacje o procesie realizacji
- Dane kontaktowe i lokalizacyjne

## Konwersja do formatu Chatfuel

Chatfuel wykorzystuje następujące formaty danych:

### 1. Intencje - pliki CSV lub JSON zawierające:

- Nazwę intencji
- Przykładowe wypowiedzi (min. 10-15 dla każdej intencji)
- Odpowiedzi

### 2. Encje - pliki CSV lub JSON zawierające:

- Nazwę encji
- Wartości encji
- Synonimy dla każdej wartości

## Struktura pliku CSV dla intencji

Plain Text

```
intent_name,utterance,response
info_pakiety,Jakie pakiety wykończeniowe oferujecie?,NovaHouse oferuje cztery
pakiety wykończeniowe: Waniliowy (podstawowy), Cynamonowy (rozszerzony),
Szafranowy (premium) i Pomarańczowy (luksusowy). Każdy pakiet zawiera różny
zakres prac i materiałów.
info_pakiety,Opowiedz mi o waszych pakietach,NovaHouse oferuje cztery pakiety
wykończeniowe: Waniliowy (podstawowy), Cynamonowy (rozszerzony), Szafranowy
(premium) i Pomarańczowy (luksusowy). Każdy pakiet zawiera różny zakres prac
i materiałów.
```

## Struktura pliku JSON dla intencji

JSON

```
{
  "intents": [
    {
      "name": "info_pakiety",
      "utterances": [
        "Jakie pakiety wykończeniowe oferujecie?",
        "Opowiedz mi o waszych pakietach",
        "Chciałbym poznać ofertę pakietów",
        "Jakie macie opcje wykończenia?",
        "Czym różnią się wasze pakiety?"
      ],
      "responses": [
        "NovaHouse oferuje cztery pakiety wykończeniowe: Waniliowy
        (podstawowy), Cynamonowy (rozszerzony), Szafranowy (premium) i Pomarańczowy
        (luksusowy). Każdy pakiet zawiera różny zakres prac i materiałów."
      ]
    }
  ]
}
```

## Struktura pliku CSV dla encji

Plain Text

```
entity_name,value,synonyms
pakiet_wykonczeniowy,waniliowy,"podstawowy,standard,ekonomiczny,najtańszy"
pakiet_wykonczeniowy,cynamonowy,"rozszerzony,średni,pośredni"
pakiet_wykonczeniowy,szafranowy,"premium,wyższy,lepszy"
pakiet_wykonczeniowy,pomarańczowy,"luksusowy,najlepszy,top,eksyluzywny"
```

## Struktura pliku JSON dla encji

JSON

```
{  
  "entities": [  
    {  
      "name": "pakiet_wykonczeniowy",  
      "values": [  
        {  
          "value": "waniliowy",  
          "synonyms": ["podstawowy", "standard", "ekonomiczny", "najtańszy"]  
        },  
        {  
          "value": "cynamonowy",  
          "synonyms": ["rozszerzony", "średni", "pośredni"]  
        },  
        {  
          "value": "szafranowy",  
          "synonyms": ["premium", "wyższy", "lepszy"]  
        },  
        {  
          "value": "pomarańczowy",  
          "synonyms": ["luksusowy", "najlepszy", "top", "ekskluzywny"]  
        }  
      ]  
    }  
  ]  
}
```

## Skrypt konwersji danych

Poniżej znajduje się przykładowy skrypt Python do konwersji istniejącej bazy wiedzy do formatu Chatfuel:

Python

```
import json  
import csv  
import os  
  
# Ścieżki do plików  
input_file = "baza_wiedzy.md"  
output_intents_json = "chatfuel_intents.json"  
output_entities_json = "chatfuel_entities.json"
```

```

# Funkcja do parsowania bazy wiedzy
def parse_knowledge_base(file_path):
    intents = []
    entities = []

    # Implementacja parsowania pliku Markdown
    # ...

    return intents, entities

# Funkcja do zapisywania intencji w formacie JSON
def save_intents_json(intents, output_file):
    data = {"intents": intents}
    with open(output_file, 'w', encoding='utf-8') as f:
        json.dump(data, f, ensure_ascii=False, indent=2)

# Funkcja do zapisywania encji w formacie JSON
def save_entities_json(entities, output_file):
    data = {"entities": entities}
    with open(output_file, 'w', encoding='utf-8') as f:
        json.dump(data, f, ensure_ascii=False, indent=2)

# Główna funkcja
def main():
    intents, entities = parse_knowledge_base(input_file)
    save_intents_json(intents, output_intents_json)
    save_entities_json(entities, output_entities_json)
    print(f"Zapisano {len(intents)} intencji i {len(entities)} encji.")

if __name__ == "__main__":
    main()

```

## Struktura bazy wiedzy w Chatfuel

### Komponenty bazy wiedzy

Baza wiedzy w Chatfuel składa się z następujących komponentów:

1. **Intencje (Intents)** - reprezentują cel lub zamiar użytkownika
2. **Encje (Entities)** - reprezentują kluczowe informacje w wypowiedziach użytkownika
3. **Konteksty (Contexts)** - zarządzają przepływem konwersacji

4. **Odpowiedzi (Responses)** - treści zwracane użytkownikowi

5. **Fulfillment** - akcje wykonywane w odpowiedzi na intencje

## Intencje dla NovaHouse

Dla chatbota NovaHouse, proponujemy następujące intencje:

1. **Powitanie (welcome)** - powitanie użytkownika

2. **Informacje o firmie (info\_firma)** - informacje o NovaHouse

3. **Pakiety wykończeniowe (info\_pakiety)** - informacje o pakietach

4. **Szczegóły pakietu (info\_pakiet\_szczegoly)** - szczegóły konkretnego pakietu

5. **Proces realizacji (info\_proces)** - informacje o procesie realizacji

6. **Cennik (info\_cennik)** - informacje o cenach

7. **Kontakt (info\_kontakt)** - dane kontaktowe

8. **Lokalizacja (info\_lokalizacja)** - informacje o lokalizacji

9. **Umówienie spotkania (booking\_spotkanie)** - umawianie spotkania

10. **Zmiana spotkania (booking\_zmiana)** - zmiana terminu spotkania

11. **Odwołanie spotkania (booking\_odwołanie)** - odwołanie spotkania

12. **Kontakt z konsultantem (contact\_konsultant)** - prosba o kontakt z konsultantem

13. **Fallback (fallback)** - obsługa nieroznalezionych intencji

## Encje dla NovaHouse

Dla chatbota NovaHouse, proponujemy następujące encje:

1. **Pakiet wykończeniowy (pakiet\_wykonczeniowy):**

- waniliowy
- cynamonowy

- szafanowy
- pomarańczowy

**2. Metraz lokalu (metraz\_lokalu):**

- mały (do 50m<sup>2</sup>)
- średni (50-100m<sup>2</sup>)
- duży (powyżej 100m<sup>2</sup>)

**3. Typ nieruchomości (typ\_nieruchomosci):**

- mieszkanie
- dom
- biuro
- lokal usługowy

**4. Miasto (miasto):**

- Gdańsk
- Gdynia
- Sopot
- Warszawa
- inne

**5. Element wykończenia (element\_wykonczenia):**

- podłogi
- ściany
- sufity
- łazienka

- kuchnia
- elektryka
- hydraulika

## Import danych treningowych

### Przygotowanie plików do importu

1. Przygotuj pliki JSON lub CSV zgodnie z formatem Chatfuel
2. Upewnij się, że kodowanie plików to UTF-8
3. Sprawdź poprawność składni JSON (jesli używasz tego formatu)

### Import przez konsolę Chatfuel

1. Zaloguj się do panelu Chatfuel Business
2. Przejdz do sekcji "AI" > "Knowledge Base"
3. Kliknij "Import"
4. Wybierz przygotowane pliki
5. Potwierdź import

### Import przez API Chatfuel

Alternatywnie, możesz użyć API Chatfuel do importu danych:

Python

```
import requests
import json

# Konfiguracja
api_key = "YOUR_API_KEY"
bot_id = "YOUR_BOT_ID"
url = f"https://api.chatfuel.com/v1/bots/{bot_id}/knowledge-base/import"
```

```
# Dane do importu
with open("chatfuel_intents.json", "r", encoding="utf-8") as f:
    data = json.load(f)

# Wysłanie żądania
headers = {
    "Authorization": f"Bearer {api_key}",
    "Content-Type": "application/json"
}
response = requests.post(url, headers=headers, json=data)

# Sprawdzenie odpowiedzi
if response.status_code == 200:
    print("Import zakończony sukcesem")
    print(response.json())
else:
    print(f"Błąd importu: {response.status_code}")
    print(response.text)
```

## Konfiguracja intencji i encji

### Konfiguracja intencji

Po imporcie, należy skonfigurować każdą intencję:

1. Przejdź do sekcji "AI" > "Intents"
2. Dla każdej intencji:
  - Sprawdź i uzupełnij przykładowe wypowiedzi (min. 15-20)
  - Skonfiguruj parametry (encje do wyodrębnienia)
  - Skonfiguruj odpowiedzi (teksty, przyciski, karty)
  - Ustaw konteksty wejściowe i wyjściowe (jesli potrzebne)
  - Skonfiguruj fulfillment (jesli potrzebny)

### Konfiguracja encji

Po imporcie, należy skonfigurować każdą encję:

1. Przejdz do sekcji "AI" > "Entities"
2. Dla każdej encji:
  - Sprawdź i uzupełnij wartości
  - Dodaj synonimy dla każdej wartości
  - Ustaw typ encji (system lub niestandardowa)
  - Skonfiguruj rozszerzanie (fuzzy matching)

## Konfiguracja kontekstów

Dla bardziej złożonych konwersacji, należy skonfigurować konteksty:

1. Przejdz do sekcji "AI" > "Contexts"
2. Utwórz konteksty dla różnych stanów konwersacji:
  - booking\_flow - dla procesu umawiania spotkania
  - pakiet\_info - dla informacji o konkretnym pakiecie
  - contact\_flow - dla procesu kontaktu z konsultantem

## Konfiguracja odpowiedzi

Dla każdej intencji, należy skonfigurować odpowiedzi:

1. Przejdz do sekcji "AI" > "Responses"
2. Dla każdej odpowiedzi:
  - Utwórz warianty odpowiedzi (min. 3-5 dla naturalności)
  - Skonfiguruj komponenty odpowiedzi (tekst, przyciski, karty)
  - Ustaw warunki wyświetlania (jeśli potrzebne)

## Testowanie i optymalizacja bazy wiedzy

## **Testowanie rozpoznawania intencji**

1. Przejdz do sekcji "AI" > "Test"
2. Wprowadź przykładowe wypowiedzi użytkownika
3. Sprawdź, czy intencje są poprawnie rozpoznawane
4. Zidentyfikuj problematyczne przypadki

## **Testowanie wyodrębniania encji**

1. Przejdz do sekcji "AI" > "Test"
2. Wprowadź wypowiedzi zawierające encje
3. Sprawdź, czy encje są poprawnie wyodrębniane
4. Zidentyfikuj problematyczne przypadki

## **Testowanie przepływów konwersacji**

1. Przejdz do sekcji "Preview"
2. Przeprowadź testowe konwersacje
3. Sprawdź, czy przepływy działają zgodnie z oczekiwaniemi
4. Zidentyfikuj problematyczne przypadki

## **Optymalizacja bazy wiedzy**

Na podstawie wyników testów, należy zoptymalizować bazę wiedzy:

1. Dodaj więcej przykładowych wypowiedzi dla problematycznych intencji
2. Dodaj synonimy dla problematycznych encji
3. Popraw odpowiedzi, które nie są satysfakcjonujące
4. Dostosuj przepływy konwersacji, które nie działają płynnie

# Zarządzanie bazą wiedzy

## Monitorowanie wydajności

1. Przejdz do sekcji "Analytics"

2. Monitoruj:

- Rozpoznane intencje
- Nierozpoznane intencje
- Najczęstsze pytania
- Skuteczność odpowiedzi

## Aktualizacja bazy wiedzy

Proces aktualizacji bazy wiedzy:

1. Regularnie analizuj nierozpoznane intencje
2. Identyfikuj nowe pytania i tematy
3. Dodawaj nowe intencje i przykładowe wypowiedzi
4. Aktualizuj odpowiedzi zgodnie z najnowszymi informacjami

## Procedura rozszerzania bazy wiedzy

1. Zbierz nowe pytania i tematy
2. Przygotuj odpowiedzi we współpracy z zespołem NovaHouse
3. Dodaj nowe intencje i przykładowe wypowiedzi
4. Przetestuj nowe intencje
5. Wdroż aktualizacje

## Najlepsze praktyki

## Intencje

- Używaj minimum 15-20 przykładowych wypowiedzi dla każdej intencji
- Używaj różnorodnych sformułowań i struktur zdań
- Unikaj nakładających się intencji
- Grupuj powiązane intencje (np. info\_pakiet\_waniliowy, info\_pakiet\_cynamonowy)

## Encje

- Używaj minimum 5-10 synonimów dla każdej wartości encji
- Włącz rozszerzanie (fuzzy matching) dla tolerancji błędów
- Używaj systemowych encji dla dat, liczb, adresów email itp.
- Definiuj własne encje dla specyficznych pojęć branżowych

## Odpowiedzi

- Przygotuj minimum 3-5 wariantów każdej odpowiedzi
- Używaj krótkiego, zwięzłego języka
- Dodawaj przyciski szybkich odpowiedzi dla najczęstszych follow-up
- Używaj rich media (zdjęcia, karty) dla lepszego doswiadczenia

## Testowanie

- Testuj z różnymi wariantami pytań
- Testuj z błędami pisowni i gramatyki
- Testuj pełne przepływy konwersacji
- Angażuj zespół NovaHouse w testowanie

Przygotował: Michał Marini

Data: 8 lipca 2025