Rozpoznawanie języka na serwerach discord

Konrad Synowiec, Michał Soboń

Projekt skupia się na identyfikacji języka, którym najczęściej posługują się użytkownicy danego serwera Discord. Aplikacja została wykonana w języku Python z wykorzystaniem bibliotek discord.py, fasttext oraz iso639-lang.

Istnieje wiele rodzajów botów, które używane są na serwerach Discord, między innymi:

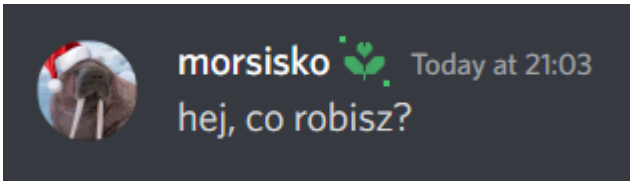
- Do moderacji (blokowania i wyciszania użytkowników gdy wyślą wiadomość niezgodną z regulaminem serwera)
- Do odtwarzania muzyki
- Zachęcające do aktywności na serwerze
- Boty pomagające nowemu użytkownikowi lepiej poznać serwer

Boty są używane na milionach serwerów, przez dziesiątki milionów użytkowników. Przykładowo bot do moderacji Dyno[1] jest obecny na prawie 8 milionach serwerów. Bot premiujący użytkowników za aktywność (poprzez zdobywanie kolejnych poziomów i przyznawanie odpowiednich ról) o nazwie mee6[2] działa na ponad 18 milionach serwerów.

Jasnym jest, że boty te muszą w jakiś sposób komunikować się z użytkownikami. Najpopularniejszym sposobem jest oczywiście wydawanie komunikatów tekstowych w języku angielskim, natomiast z Discorda korzystają ludzie różnych narodowości mówiący w różnych językach, także dzieci, które niekiedy nie znają żadnego języka obcego. Twórcy botów chcąc trafić do jak największej liczby użytkowników muszą sprawić, by wydawane przez boty komunikaty były zrozumiane przez ludzi. Jak wiadomo bardzo trudno jest przetłumaczyć komunikaty na wszystkie języki świata, dlatego dobrym pomysłem byłoby sprawdzić, jakie języki są najczęściej używane na serwerach, na których działa nasz bot.

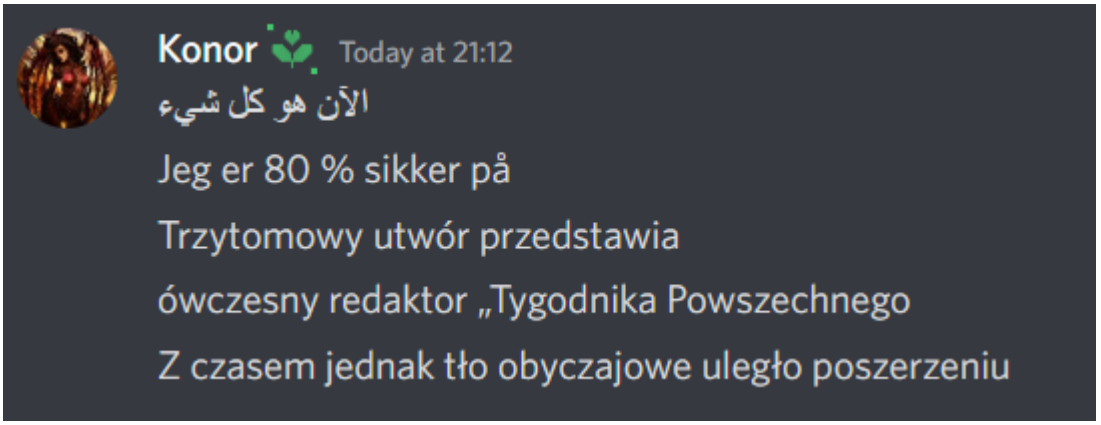
Nasz program na bieżąco zbiera informacje o tym w jakim języku napisane zostały wiadomości. Następnie wyznaczany jest język, w którym napisane zostało najwięcej wiadomości na danym serwerze. Algorytm rozpoznawania języka na serwerze działa w oparciu o bibliotekę fasttext w następujący sposób: Dla każdej wysłanej wiadomości wybierane jest pięć najbardziej prawdopodobnych języków. Prawdopodobieństwo jest określane liczbą rzeczywistą z przedziału od 0 do 1. Następnie te prawdopodobieństwa sumowane są do ogólnych wyników danego serwera, dzięki czemu można wyłonić najpopularniejszy język. W celu zwiększenia autentyczności wskazań nie procesujemy wiadomości krótszych niż dziesięć

znaków, oraz krótszych niż trzy wyrazy, gdyż im krótszy tekst tym mniejsze prawdopodobieństwo jego poprawnego rozpoznania.




```
Message from morsisko#9507: hej, co robisz?
I assume the message is written in Polish. I'm 95.99% sure
```

Program poprawnie zidentyfikował język wiadomości jako polski.




```
Message from Konor#5883: □□□□ □□ □□ □□□
I assume the message is written in Arabic. I'm 99.97% sure
Message from Konor#5883: Jeg er 80 % sikker på
I assume the message is written in Danish. I'm 99.79% sure
Message from Konor#5883: Trzytomowy utwór przedstawia
I assume the message is written in Polish. I'm 100.0% sure
Message from Konor#5883: ówczesny redaktor „Tygodnika Powszechnego
I assume the message is written in Polish. I'm 97.38% sure
Message from Konor#5883: Z czasem jednak tło obyczajowe uległo poszerzeniu
I assume the message is written in Polish. I'm 98.87% sure
```

Przy wyłączaniu bota wypisywane są zapisane wcześniej statystyki, czyli przeważający język dla każdego serwera na którym jest bot:

```
Top language used in sever 1044955483041169488: Polish
```

Przewidywania na temat wszystkich serwerów są zapisywane do pliku predictions.json tak, by po restarcie bota nie zostały utracone zebrane statystyki.


```
{
    "10449554830411694 88": {
        "English": 0.9547320711080829,
        "Chinese": 0.0031975363381206 99,
        "Russian": 0.0075732055502157 89,
        "Thai": 0.0027171839028596 88,
        "Persian": 0.0024319542571902277,
        "Arabic": 2.9990036190411049,
        "Egyptian Arabic": 0.0008977038378361613,
        "Spanish": 0.0003362446477694 9026,
        "Ukrainian": 0.0043818745980388489,
        "Urdu": 0.0000603467124165 0358,
        "Polish": 9.800054907798767,
        "Esperanto": 0.076213939037188 54,
        "Hungarian": 0.0435878317803144 48,
        "Indonesian": 0.0022559707285836 34,
        "Danish": 1.9958807229995728,
        "Norwegian": 0.0042586191557347778,
        "Norwegian Nynorsk": 0.0000375589261238 9289,
        "Swedish": 0.0000226110350922 6814,
        "German": 0.0140365181396191 48,
        "Lithuanian": 0.0090916472545359 29,
        "Croatian": 0.0101112201809883 12,
        "Czech": 0.0054894564673304 56,
        "Slovenian": 0.0003095374559052 2885
    }
}
```

```python
async def on_message(self, message):
    # Getting server id
    server_id = str(message.guild.id)

    # Predict top 5 languages using fasttext pretrained model
    message_predictions = predict(message.content)

    # Print author and content of the message
    print(f'Message from {message.author}: {message.content}')

    # If the message is too short stop processing
    if len(message.content) < 10 or len(message.content.split()) < 3:
        print("This message is too short to process.")
        return

    # If it's first message in this server - create empty dict
    if server_id not in servers:
        servers[server_id] = {}

    # Var for storing best language match for this message
    best_lang = None

    # Loop over top 5 predicted languages
    for lang in message_predictions:
        if best_lang is None:
            best_lang = lang

        # If current lang has got higher probability than previous best,
        # make the current language the best one
        if message_predictions[lang] > message_predictions[best_lang]:
            best_lang = lang

        # Add probability of prediction for particular language to server statistics.
        if lang not in servers[server_id]:
            servers[server_id][lang] = message_predictions[lang]
        else:
            servers[server_id][lang] += message_predictions[lang]

    print("I assume the message is written in {}. I'm {}% sure".format(best_lang,\
    round(message_predictions[best_lang] * 100, 2)))
```

Powyżej znajduje się istotny fragment programu, który procesuje każdą nową wiadomość nadesłaną przez użytkownika. Jest on opatrzony komentarzami celem łatwiejszego zrozumienia kodu.

Wnioski:
We wszystkich naszych testach biblioteka fasttext okazała się nieomylna. Za każdym razem poprawnie rozpoznaje język, w którym została napisana wiadomość. Tym samym twórcy botów mogliby rzeczywiście zaimplementować proponowane przez nas rozwiązanie problemu i dowiedzieć się, jakie języki są używane przez ludzi, którzy korzystają z ich produktu.

Bibliografia:
[1] - https://dyno.gg/
[2] - https://mee6.xyz/pl/