beCP
2023
Taak 2.1: Vertaaltraining (translation)

Auteur: het beCP team Voorbereiding: Robin Jadoul

Opmerking: Dit is een output only *taak. Dit wil zeggen dat je enkel de resultaten van je berekeningen moet indienen, en niets van code. Er zijn verschillende input bestanden waarvoor je oplossing tesamen of apart kan indienen. Natuurlijk wil dit formaat ook zeggen dat je niet verplicht of verondersteld wordt hetzelfde programma voor elke input bestand te gebruiken. Daarnaast kan je misschien ook proberen of je sommige dingen met de hand kan oplossen.*

Je bent aan het training om vertaler te worden, zodat je de lokale taal kan spreken op internationale programmeerwedstrijden. Het huiswerk is meestal heel veel werk, en je zou je tijd liever spenderen aan training voor het programmeeraspect van de wedstrijden. Dus wat je normaal doet is gewoonweg alles aan een vertaalprogramma op basis van artificiële intelligentie geven.

Maar oh nee! Je internetverbinding valt weg, en je moet je huiswerk over drie uur al indienen. Jouw taak is om een hele hoop documenten de classifiëren, en voor elk document aan te geven in welke taal het geschreven is, op basis van een paar voorbeelden per taal.

De lijst van talen waar je mee bezig bent is de volgende:

- English (`en`)
- Esperanto (`eo`)
- Māori (`mi`)
- Bahasa Indonesia (`id`)
- lojban (`jbo`)

Input

In de download voor deze taak vind je 5 mappen. In de eerste map, `reference`, vind je de gekende voorbeelden per taal, met de naam `[taal]_[i].txt`. In elk van de andere mappen `set_[subtaak]` vind je een aantal (F) bestanden, enkel met de naam `[i].txt`. Elk van deze bestanden is een document warrovoor je de correcte taal moet bepalen.

Output

Voor elk van de vier mappen dien je een enkel bestand in. De i-de lijn van het bestand dient een enkel woord te bevatten: de (2 or 3-letterige) taalcode voor de taal in het i-de bestand. Er is al wat skeletcode voorzien om van de bestanden te kunnen lezen.

Scoring

Voor een verzameling bestanden (een subtaak) waar F bestanden geclassifieerd moeten worden, als je x van de F bestanden correct classifieert, krijg je $25 \cdot \frac{x}{F}$ punten.

Bijkomende beperkingen

Subtaak	Punten	Beperkingen
A	25	$F = 20$
B	25	$F = 100$
C	25	$F = 1\,000$
D	25	$F = 5\,000$

Je code uitvoeren

In de downloads voor deze taak vind je skeletcode terug in `translation.cpp`. Daar kan je de code invullen om de taal van een stuk tekst te bepalen. Dit doe je door cd functie `string classify(string)` te implementeren, die de inhoud van een bestand als input krijgt, en een van de vijf mogelijke taalcodes als string als resultaat geeft. Een functie `bool contiains(string haystack, string needle)` is ook voorzien als voorbeeld of iets dat je met je code kan doen. Je mag deze functie gebruiken, maar dat is niet verplicht.

Verder kan je ook het bestand `driver.cpp` vinden. Dit bestand voorziet de `main` functie voor het programma, en zorgt er voor dat je `classify` functie uitgevoerd kan worden voor een enkel bestand of voor alle bestanden in een folder, in de correcte volgorde.

Om een programma te compileren, gebruik je het volgende commando:
`g++ -std=c++11 -Wall -Wextra -Wshadow translation.cpp -o translation`.
om je code op een enkel bestand uit te voeren, bijvoorbeeld `reference/en_1.txt`, kan je dit commando gebruiken
`./translation reference/en_1.txt`
en om het op elk bestand in bvb de `set_A` folder uit te voeren:

`./translation set_A`. Deze commandos tonen de talen die je programma detecteert op het scherm, en waarschuwen je voor onbekende taalcodes. Om deze output naar een bestand op te slaan zodat je het kan indienen, kan je `> bestand.txt` aan het eind van het commando toevoegen, als volgt: `./translation set_A > oplossing_set_A.txt`.

Als je zelf debug informatie wil printen, zorg er dan zeker voor om `cerr` te gebruiken in plaats van `cout`.

Je mag om technische hulp vragen om je code uit te kunnen compileren en uitvoeren om de data van de taak, door het aan een toezichter te vragen of door een vraag in CMS te stellen.