**Documentatie letterfrequenties**

In dit bestand staat de documentatie voor de opdracht letterfrequenties. Het doel van de opdracht is om een algoritme te implementeren die een bestand met engelse en nederlandse zinnen kan classificeren. Dit moet in de vorm van een map reduce architectuur zodat het schaalbaar is.

**Het programma**

Ik heb ervoor gekozen om het algoritme als volgt op te zetten:

Eerst creëer ik een lijst met alle letters van het alfabet, een spatie en een % voor tekens die niet in het alfabet voorkomen. (get\_alphabet\_array())

De eerste mapper heeft als input een tekstbestand. Het tekstbestand wordt vervolgens lijn voor lijn gelezen. De lijn wordt opgesplists in bigrammen, of te wel letterfrequenties en elke letterfrequenties krijgt een 1 toegevoegd. De zin ‘Aa bB cCc’ wordt dan:

aa, 1

a , 1

b, 1

bb, 1

b , 1

c, 1

cc, 1

cc, 1

c , 1

De eerste reducer reduced deze bigrammen en telt overeenkomende bigrammen bij elkaar op, in het geval van de vorige zin dus:

aa, 1

a , 1

b, 1

bb, 1

b , 1

c, 1

cc, 2

c , 1

De tweede mapper heeft als input de gereducede bigrammen. Het doel van deze mapper is het aantal keys te tellen om de probability van bigrammen in een volgende stap te kunnen berekenen. Stel dat de key ‘a’ 100 keer voorkomt en de letterfrequentie ‘aa’ 10 keer, dan is de probability van aa dus 10%. In deze stap worden de keys dus geteld. De reducer die hierop volgt reduced vervolgens de aa, value, ab value etc weer naar hoe vaak een key met de start a voorkomt, en dat dan natuurlijk voor alle letters.

De derde mapper heeft twee inputs, de gereduceerde bigrammen van de eerste reducer en de gereduceerde key values van de tweede reducer. Het enige wat deze mapper doet is de counts van de eerste reducer / de totale count om een probabilty van een bigram te berekenen. De bijbehorende reducer zet de berekende probabilities in een dict.

Het eindproduct is dus een matrix met bigram probabilities. Deze map reduce structuur is dus bruikbaar voor zowel de engelse als nederlandse tekst. Zo kan makkelijk een engelse en nederlandse matrix opgebouwd worden met berekende probabilities -> of te wel onze classifier!

De volgende stap is het uitvogelen of de classifier een beetje werkt. Hiervoor worden de mixed\_sentences tekst file en de getrainde engelse en nederlandse matrixen gebruikt.

de language mapper doet in principe hetzelfde als de eerste mapper, er wordt een tekst ingelezen, samen met de getrainde engelse en nederlandse matrixen. De tekst wordt opgesplitst in bigrammen per zin en vervolgens wordt er per bigram bekeken wat de bijbehorende probabilities zijn voor engels en voor nederlands. Dit wordt dan weer gereduced naar counts per zin.

De laatste mapper leest de counts per zin in en kijkt welke probability hoger is. bij bijvoorbeeld 2 geclassificeerde engelse zinnen en 3 geclassificeerde nederlandse zinnen is de output als volgt:

english, 1

english, 1

dutch, 1

dutch, 1

dutch, 1

in de laatste reduce worden de counts opgeteld. Voor de mixed\_sentences tekst heb ik het volgende resultaat behaald:



Een kleine afwijking van de bedoelde 119 engelse en 73 nederlandse zinnen, maar des al niet te min een goede uitkomst.