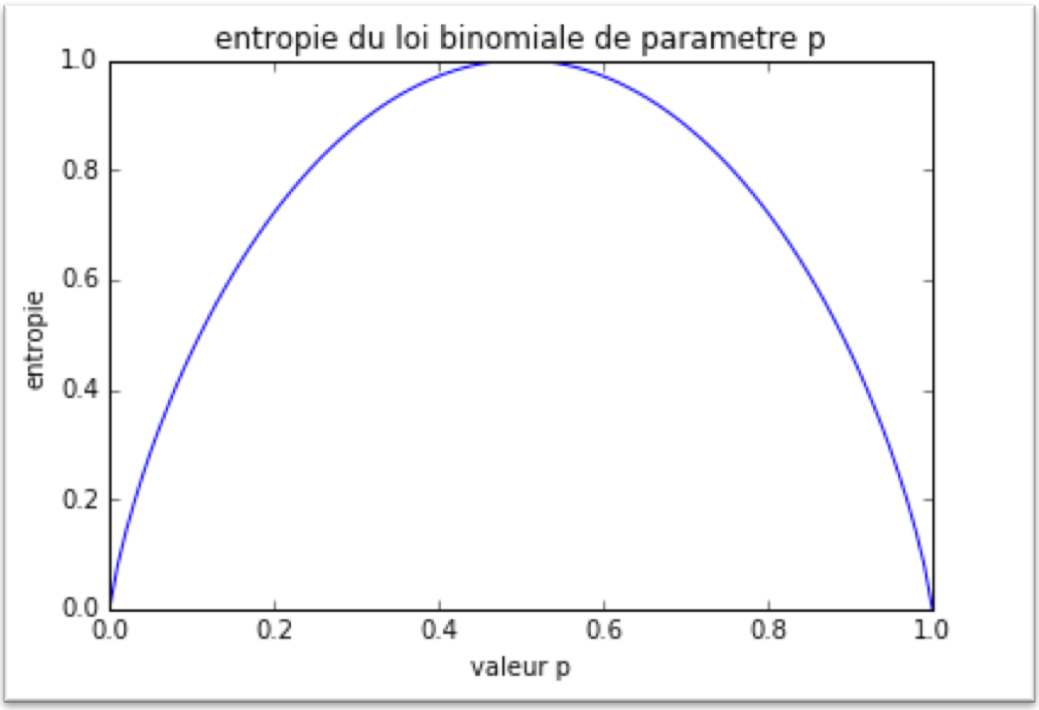
XIANG
Yangfan
3300401

Rapport du projet 1

I. Principes introductifs et codage



On observe la valeur d'entropie atteint le maximum pour p=0.5 et décroit vers les deux extrémités. Le fait qu'une entropie élevée signifie un aléa plus important, qui correspond bien au valeur de p car pour p=0.5 la probabilité qu''un évènement arrive est n'arrive pas est la même. Tandis qu'en rapprochant des deux extrémités on aura plus d'information de la réalisation ou non de l'événement donc moins d'aléa.

II. Entropie d'une langue

1) Expérience 1

Premièrement le texte considérer a subi un prétraitement qui permet de convertir les caractères spéciaux en UTF- 8 en un simple codage ASCII, transformer les majuscules en minuscules, supprimant les sauts de lignes et les tabulations. Lors de ma construction de dictionnaire de Ngrams, Pour simplifier le model, je considère que chaque mot est indépendant des autres donc de prendre en considération les espaces entre les mots. Certes il peut avoir des relations entre les mots mais dans notre cas on peut ne pas les prendre en compte.

Tableau d'entropie en fonction de la langue et de N

LANGUE	N=1	N=2	N=3	N=4	N=5	N=6
FRANÇAIS	4.0093	7.4186	10.4048	12.8663	14.7793	16.1754
ANGLAIS	4.1797	7.7540	10.7931	13.1394	14.7444	15.7732
ESPAGNOL	4.0466	7.4677	10.5086	12.9936	14.7944	15.9471

On observe que l'entropie de chaque langue est différente des autres mais ont des écartes d'ordre de dixième, donc si chaque langue est caractérisée par sa valeur d'entropie alors il faut peut-être aller voire la valeur au millième près pour déterminait la langue et en plus les entropies sont très sensibles au longueur du texte même si elles appartiennent au même langage.

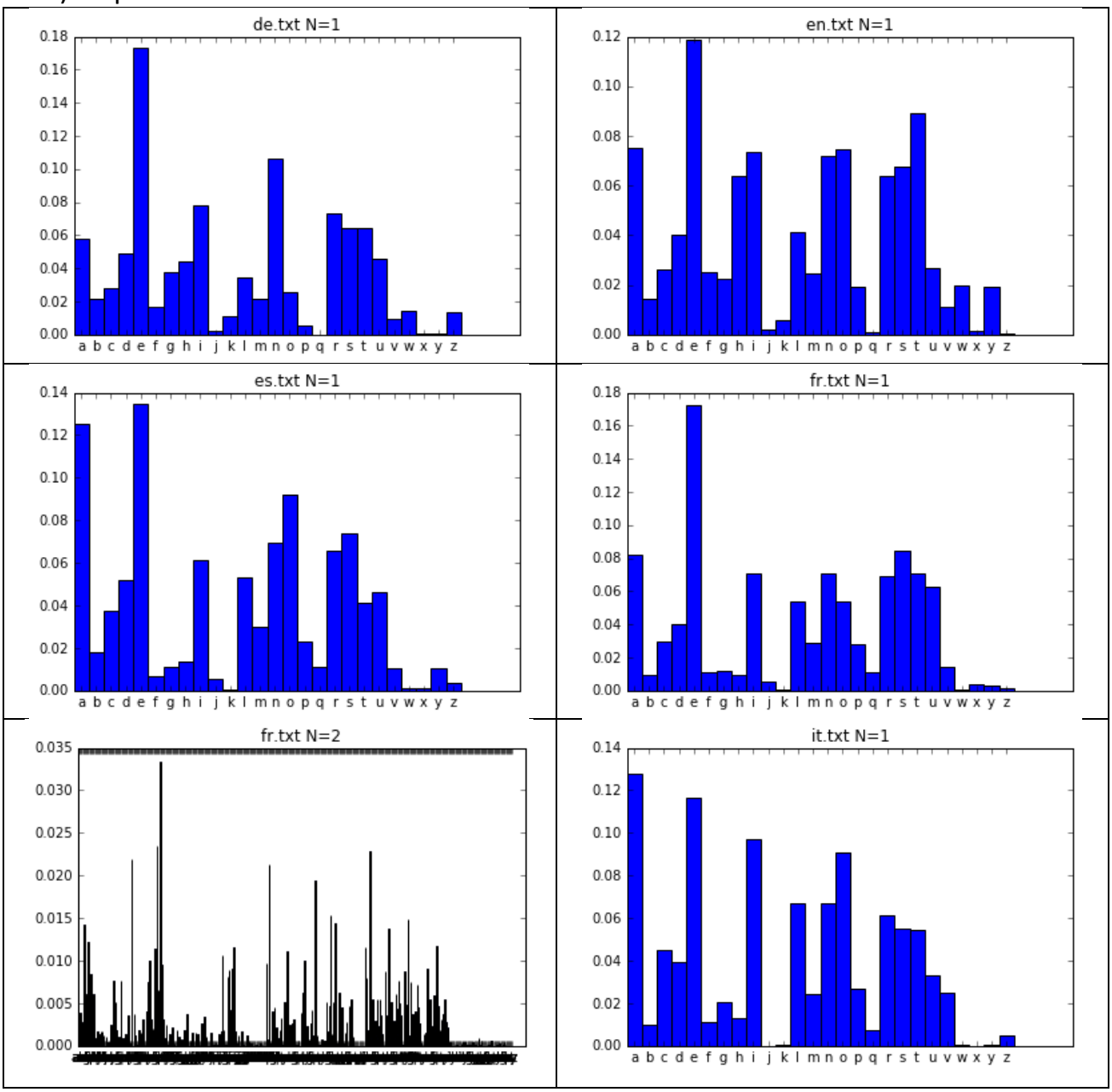
2) Expérience 2

D'après mes expériences, l'utilisation de la méthode d'entropie relative est une méthode fiable pour déterminer la langue d'un texte. Je me suis poser la question qu'est-il possible de déterminer la langue du sujet de notre projet qui ne fait que 5 pages, le résultat me montre qu'il arrive bien à déterminer la langue. Mais en poussant plus loin l'expérience, réduisant la taille du texte en un unique paragraphe de quelque phrase, on arrive toujours à déterminer mais les valeurs des entropies relatives sont très proche donc on peut en pensé que le résultat de la détermination sur un texte très court n'est pas fiable à 100%.

Dans le cas extrême de déterminer la langue qu'un mot appartient, la méthode d'entropie relative n'est plus du tout fiable, l'information contenu dans un seul mot n'est pas suffisant pour que cette méthode puisse déterminer la langue.

On en conclue que dans le cas ou les données d'entré sont suffisamment grande, la méthode d'entropie relative arrive parfaitement à déterminer la langue mais sur des tailles de donnée petite cette est hors de porté.

3) Expérience 3



Pour les Ngrams de longueur N=1, le fait de tracer la distribution (26 lettres de l'alphabet) sous forme d'histogramme nous donne une vision plus direct sur la distribution des lettres pour différente langue. Du première coup d'œil, on peut s'informer quelles sont les lettres les plus fréquents dans cheque langue. Mais pour une valeur de N à partir de 2, l'histogramme devient illisible car on aura 26^N lettre de longueur N à afficher, on a donc perdu l'avantage d'avoir une représentation visuel des caractères de la langue.

III. Classification bayésienne

1) Partie théorique

On a comme information un mot et on veut déterminer la langue auquel il appartient donc la probabilité de la langue sachant le mot $p(l|w)$ .D'après la formule de Bayes on a $p(l|w) = \frac{p(w|l)p(l)}{p(w)}$, $p(w|l)$ la probabilité d'apparition du mot dans la langue, $p(l)$ la probabilité de la langue (par exemple dans le monde informatique c'est l'anglais qui domine) et $p(w)$ la probabilité d'apparition du mot.
En appliquant sur l'ensemble des langues, nous arrivons à déterminer dans quelle langue le mot appartient avec une probabilité maximum.
Comme $p(w)$ est constant donc dans notre cas il n'est pas nécessaire de calculer cette valeur.
Dans mes calcules, je considère que nous trouvons dans un cas général sans plus d'information sur l'apparition des langues. Dans ce cas, il est préférable que la probabilité des langues sont uniformes donc aussi il n'est pas nécessaire de calculer cette valeur, mais si on a plus d'informations nous pouvons prendre en compte et d'ajouter au calcule.

Pour calculer $p(w|l)$ on peut décomposé w en une suite de lettre $w = w_1 w_2 \dots w_n$, considérons que les lettres sont indépendants entre eux pour simplifier le modèle on obtient $p(w|l) = p(X_1 = w_1|l)p(X_2 = w_2|l) \dots p(X_n = w_n|l)$ ce qui n'est pas le cas dans le monde réel car prenons l'exemple du français ou après la lettre q apparaît une lettre u.
On peut améliorer notre modèle en en supposant une dépendance d'ordre m, on obtient $p(X_i|X_{i-m} \dots X_{i-1}, l) = \frac{p(X_{i-m} \dots X_{i-1}, l|X_i)p(X_i)}{p(X_{i-m} \dots X_{i-1}, l)}$ or on a fait l'hypothèse que les lettres sont indépendants entre eux donc la formule devient $\frac{p(X_{i-m} \dots X_i, l)}{p(X_{i-1} \dots X_{i-m}, l)}$.
Finalement on applique la fonction log pour que la différence des valeurs soit plus flagrante.

2) Partie expérimentale

En utilisant la méthode bayésienne sans dépendance (donc nbDep=0), l'algorithme nous retourne bien la langue que le texte est écrit pour des tailles de texte pas très petite (supérieur au paragraphe). Mais si on restreint le texte à un unique mot, la méthode bayésienne a le même défaut que la méthode d'entropie. Les données contenues dans un seul mot ne sont pas suffisant pour déterminer la langue.

```
fr.txt  -23.44026083276998
en.txt  -23.177191419887016
es.txt  -23.23218176656562
fr4c.txt appartient au en.txt avec un valeur
bayesienne en log de -23.177191419887016
```

Si on introduit la dépendance dans, on observe une très grande amélioration sur la détermination de la langue qui est traduit par l'écarte entre la valeur retourner par l'algorithme de la langue auquel le mot appartient et les autres langues. Après multitude d'essai, pour une dépendance d'ordre de comprise entre 3 et 6, et des lettres (w_i) formées que d'une seul lettre d'alphabet, la qualité (fiabilité) du résultat est optimal. Sans doute, l'explication est qu'il y a plus de mot de taille voisinant entre 3 et 6, et que les mots sont composés d'une seul lettre d'alphabet.

```
In [77]: deterLangBaye("fr2.txt",["fr.txt", "en.txt", "es.txt"], 4, 1)
fr.txt  -13803169.466956703
en.txt  -53561196.21160371
es.txt  -48146016.89174957
fr2.txt appartient au fr.txt avec un valeur bayesienne en log de
-13803169.466956703
```

(Pour les probabilités 0, je considère que sa valeur en log est de -100 qui correspond log(7.88e-31) pour pouvoir comparait dans le cas ou on a décidé d'applique le log pour amplifier l'écart entre les résultats des probabilités)

IV. Codage de Huffman

	fr.txt	en.txt	de.txt	it.txt	es.txt
Esperance(bit/lettre)	4.0513	4.2074	4.0946	4.0342	4.0719

Si les 26 lettres de l'alphabet sont équiprobable, alors on aura besoin de $log_2(26)bit = 4{,}70$ soit 5 bits pour codé les 26 lettres. Tandis que si la probabilité d'apparition des lettres n'est pas équiprobable, ce qui est le cas dans les langues en réel, en appliquant le codage de Huffman on peut diminuer le nombre de bit totale pour codé un texte.
En appliquant le codage de Huffman sur les textes, j'ai calculé l'espérance pour codé une lettre, le résultat se trouve dans le tableau récapitulatif ci-dessus. On observe qu'on en gagne ≈ 0.6 bit par lettre.