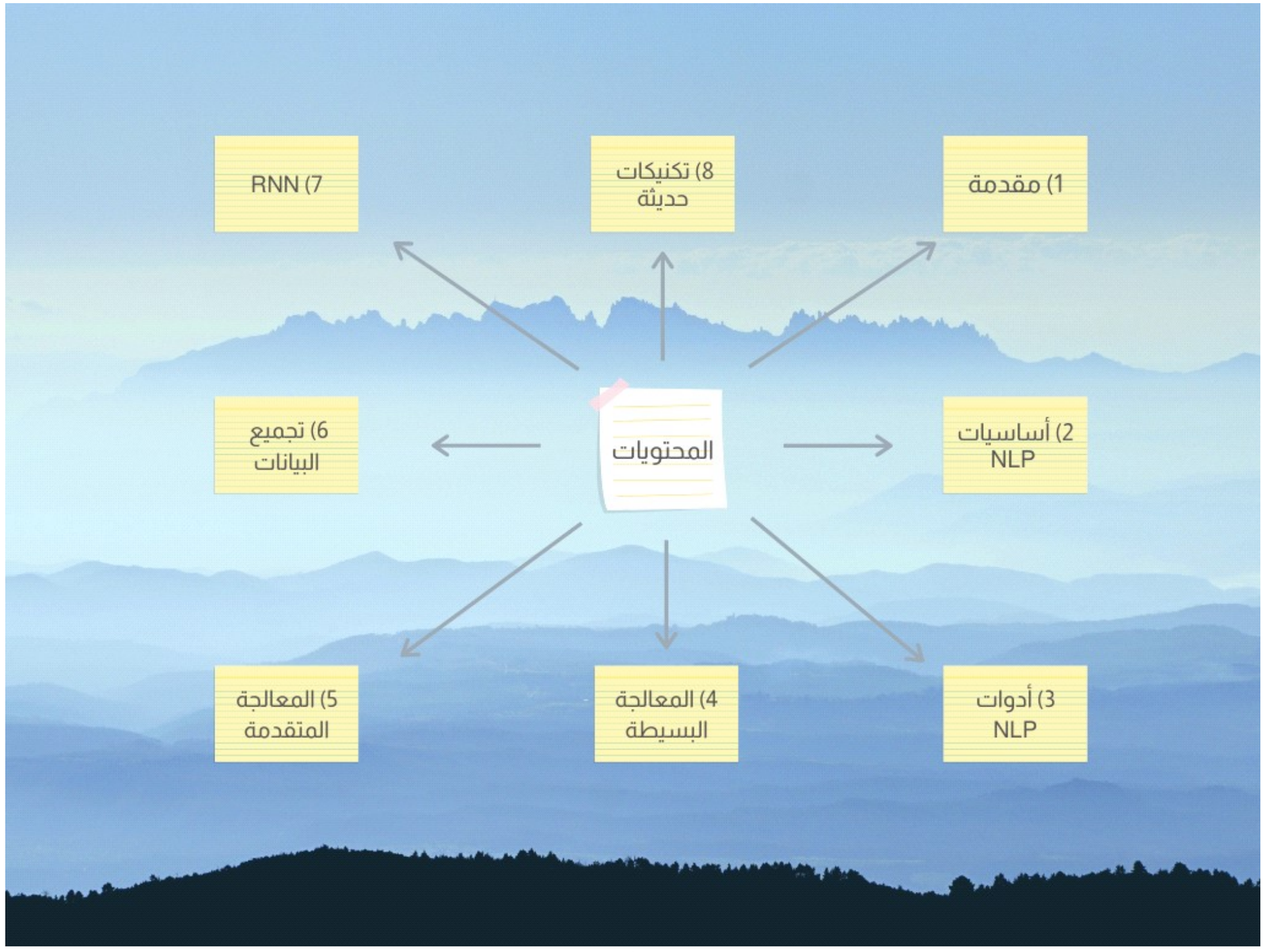
Natural Language Processing المعالجة اللغوية الطبيعية



المحتويات

القسم الخامس : المعالجة المتقدمة للنصوص

الجزء الثالث عشر : التصحيح التلقائي Auto Correct

==========

و هي الأداة الخاصة بتصحيح الكلمات , واختيار أقرب كلمة مناسبة , الي الكلمة الخاطئة التي تمت كتابتها . .

فلو قمت بكتابة :

Happy birth day my deah friend

فيجب ان يتم تصحيح الكلمة الي dear لانها هي المطلوبة هنا

و يمر التصحيح بأربع مراحل :

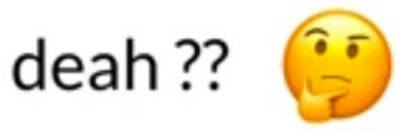
1. **تحديد هل الكلمة خطأ أم لا**
2. **تحديد الحرف الخاطئ , و تحديد الكلمات البديلة**
3. **فلترة الكلمات المناسبة**
4. **اختيار افضل كلمة**

_

1. **تحديد هل الكلمة خطأ أم لا**

و ذلك عبر مقارنة هذه الكلمة مع القاموس الموجود لهذه اللغة , لتحديد هل هذه الحروف تمثل كلمة صحيحة ام لا

```
if word not in vocab:
    misspelled = True
```



2. تحديد الحرف الخاطئ , و تحديد الكلمات البديلة

يتم هذا باستخدام التباديل و التوافيق , عبر تغيير كل حرف من الحروف و اختيار موضعها جميع حروف الابجدية , لعمل قائمة بكل الكلمات المقترحة , كما اننا احيانا نضيف عدد من الحروف الاضافية بفرض ان الكلمة المكتوبة اقل من الكلمة الحقيقية , او تبديل الحروف , ثم حذف جميع الكلمات التي ليست في قاموس اللغة و ليس لها معني

Find strings n edit distance away

Edit: an operation performed on a string to change it

Insert	(add a letter)	'to': 'top', 'two' ...
Delete	(remove a letter)	'hat': 'ha', 'at', 'ht'
Switch	(swap 2 adjacent letters)	'eta': 'eat', 'tea'
Replace	(change 1 letter to another)	'jaw': 'jar', 'paw', ...

Given a string find all possible strings that are n edit distance away using

- Input
- Delete
- Switch
- Replace

deah

_eah

d_ar

de_r

... *etc*

3. فلترة الكلمات المناسبة

و يتم هذا عبر فلترة باقي الكلمات المتبقية , والتي لها معني , لجعلها short list للاختيار

4. اختيار افضل كلمة

و هذا عبر اختيار اقرب اوزان للكلمة المختارة عبر تضمين الكلمات , و علاقتها بمعاني باقي الكلمات في الجملة

و يتم حساب الاوزان باكثر من طريقة , منها حساب احتمالية تواجد هذه الكلمة في المواضيع المتشابهة , فلو كانت الجملة التي يتم كتابتها في الاقتصاد , فيتم حساب احتمالية تواجدها في المواضيع الاقتصادية و هكذا

Example: “I am happy because I am learning”

$$P(w) = \frac{C(w)}{V} \qquad P(\text{am}) = \frac{C(\text{am})}{V} = \frac{2}{7}$$

$P(w)$ Probability of a word

$C(w)$ Number of times the word appears

V Total size of the corpus

Word	Count
I	2
am	2
happy	1
because	1
learning	1

Total : 7

_

كما ان هناك مراحل تتم في خوارزميات اخري . .

- أولا مرحلة اكتشاف الخطأ نفسه
- ثم عمل التصحيح , والذي قد يكون :
  - تصحيح تلقائي : يقوم الموديل بتصحيحه دون سؤالك , وهي في حالة كانت الكلمة خطأ تماما و قريب من كلمة صحيحة متداولة
  - عرض اقتراح : ان يقوم الموديل بعمل اقتراح لك ان الكلمة الصحيحة هي كذا
  - عرض قائمة اقتراحات : في حالة ان هناك أكثر من كلمة قريبة مما كتبت

_

و هناك انواع من التصحيح :

- الكلمة أصلا خطأ : وهي في الكلمات التي لا تتواجد في القاموس من الاساس
- المعني خطأ : وهي في الكلمات التي موجودة في القاموس لكن استخدامها خطأ هنا I`ll go three و المفروض there

و هذه ارقام توضح مدي تواجد الخطأ و تصحيحه

26%: Web queries Wang *et al.* 2003

13%: Retyping, no backspace: Whitelaw *et al.* English&German

7%: Words corrected retyping on phone-sized organizer

2%: Words uncorrected on organizer Soukoreff &MacKenzie 2003

1-2%: Retyping: Kane and Wobbrock 2007, Gruden et al. 1983

_

و يتم اكتشاف الكلمة الخاطئة بأحد طريقتين :

- كلمة الخاطئة أصلا : عبر البحث هل الكلمة موجودة في القاموس ام لا
- المعني الخطأ : استخدام احد الخوارزميات لبحث هل المعني سليم ام لا , مثل bigram

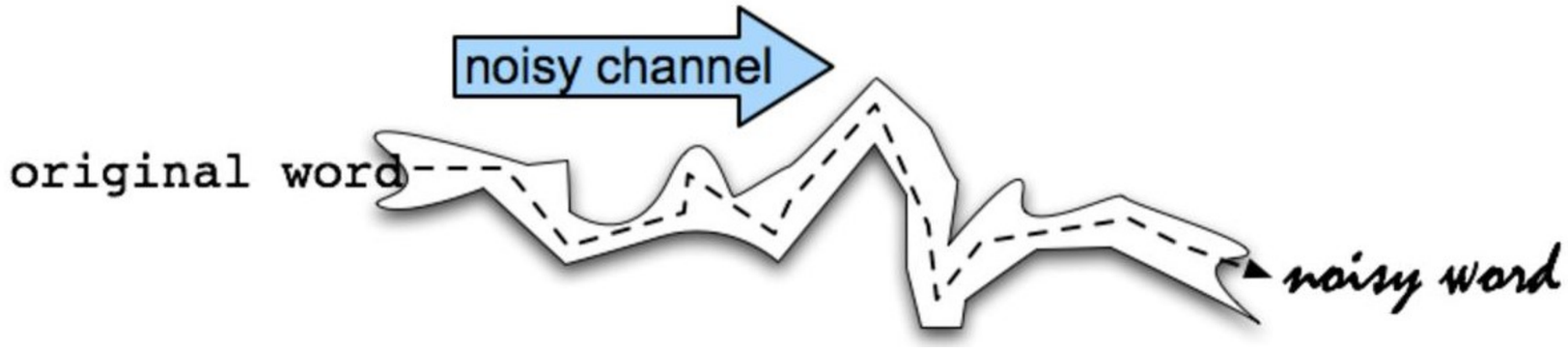
و أحيانا يتم تطبيق هذا الأمر علي كل الكلمات , لأن الكلمة الاصلية قد تكون هي ايضا غير سليمة

_

نتكلم عن النوع الاول من الاخطاء , وهو الذي يكون من كتابة كلمة لا اصل لها techer مثلا

نتناول الان أحد تكنيكات تصحيح الأخطاء , والمسمي Noisy Channel

و الذي يقوم علي فكرة أن الكلمات تدخل في مسار محدد و المسمي بنفس الاسم , لتحديد الكلمة المناسبة لها



و يتم هذا عبر حساب احتمالية الحرف الذي يتم تصحيحه , مضروبا في احتمالية وجود الكلمة نفسها , وتجريب هذا الامر في عدد من التصحيحات المحتملة للعثور علي التصحيح المناسب

مع التأكيد علي ان x تشير للتصحيح w , الكلمة نفسها

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(w \mid x)$$

$$= \underset{w \in V}{\operatorname{argmax}} \frac{P(x \mid w) P(w)}{P(x)}$$

$$= \underset{w \in V}{\operatorname{argmax}} P(x \mid w) P(w)$$

فلو كان لدينا مثلا كلمة خطأ هي acress , فنقوم أولا بالبحث عن الكلمات التي تتشابه معها في الحروف , او تتشابه في النطق

فكلمتي sea , C يتشابهان في النطق مع عدم تشابه حروفهما

ثم نقوم بحساب المسافة بين كل كلمة فيهم و الكلمة الاصلية , هذه المسافة معتمدة علي عدد التعديلات التي ستتم في الكلمة , سواء بالإضافة او الحذف او التعديل او تبديل حرفين متتاليين taecher , teacher

فهذه التعديلات للكلمات المقترحة لكلمة acress

Error	Candidate Correction	Correct Letter	Error Letter	Type
acress	actress	t	-	deletion
acress	cress	-	a	insertion
acress	caress	ca	ac	transposition
acress	access	c	r	substitution
acress	across	o	e	substitution
acress	acres	-	s	insertion
acress	acres	-	s	insertion

مع العلم ان 80% من الكلمات تم تصحيحها بخطوة واحدة , وتقريبا 100% يتم عبر خطوتين

و الا ننسي ايضا ان المسافة و الحروف الخاصة يتم وضعها في الحسبان

ثم يتم الاستعانة بال Unigram لمعرفة مدي تواجد هذه الكلمات في ال corpus علي اعتبار ان الكلمة ذات نسبة التواجد الاكبر قد تكون هي الأعلى في احتمالية ان تكون الكلمة الصحيحة

و نري هنا ان كل كلمة لها احتمالية , وهي قسمة تكرارها علي العدد الكلي لكلمات العينة (404 مليون)

word	Frequency of word	P(word)
actress	9,321	.0000230573
cress	220	.0000005442
caress	686	.0000016969
access	37,038	.0000916207
across	120,844	.0002989314
acres	12,874	.0000318463

كما انه يمكن تدريب الموديل لمعرفة مدي تكرار تصحيح حرف مكان حرف معين , و هذا علي الاربع انواع الاساسية :

- الحذف : ان يكون مكتوب xy و المفروض x : الكلمة الصحيحة هي : teacher و الخطأ : tieacher
- الإضافة: ان يكون مكتوب x و المفروض xy : الكلمة الصحيحة هي : teacher و الخطأ : techer
- التعويض: ان يكون مكتوب y و المفروض x : الكلمة الصحيحة هي : teacher و الخطأ : tiacher
- التبديل : ان يكون مكتوب xy و المفروض yx : الكلمة الصحيحة هي : teacher و الخطأ : taecher

و هنا يتم حساب p(x/w) و هي احتمالية التعديل (حذف , اضافة , تعويض , تبديل) مقسومة علي عدد الكلمة الاصلية

$$P(x|w) = \begin{cases} \frac{\text{del}[w_{i-1}, w_i]}{\text{count}[w_{i-1} w_i]}, & \text{if deletion} \\ \frac{\text{ins}[w_{i-1}, x_i]}{\text{count}[w_{i-1}]}, & \text{if insertion} \\ \frac{\text{sub}[x_i, w_i]}{\text{count}[w_i]}, & \text{if substitution} \\ \frac{\text{trans}[w_i, w_{i+1}]}{\text{count}[w_i w_{i+1}]}, & \text{if transposition} \end{cases}$$

و هنا نقوم بتطبيق القاعدة لحساب نسبة تصحيح حرف c الي ct و هكذا

Candidate Correction	Correct Letter	Error Letter	x|w	P(x|word)
actress	t	-	c|ct	.000117
cress	-	a	a|#	.00000144
caress	ca	ac	ac|ca	.00000164
access	c	r	r|c	.000000209
across	o	e	e|o	.0000093
acres	-	s	es|e	.0000321
acres	-	s	ss|s	.0000342

ثم تطبيق القانون الاصلي , لمعرفة احتمالية تواجد هذه الكلمة , عبر ضرب الرقمين في بعضهما , ويتم ضرب الرقم في مليار حتي يكون واضحا

Candidate Correction	Correct Letter	Error Letter	x|w	P(x|word)	P(word)	10^9 *P(x|w)P(w)
actress	t	-	c|ct	.000117	.0000231	2.7
cress	-	a	a|#	.00000144	.000000544	.00078
caress	ca	ac	ac|ca	.00000164	.00000170	.0028
access	c	r	r|c	.000000209	.0000916	.019
across	o	e	e|o	.0000093	.000299	2.8
acres	-	s	es|e	.0000321	.0000318	1.0
acres	-	s	ss|s	.0000342	.0000318	1.0

لكن النتيجة ليست ايجابية دائما , فكلمة actress المختارة ليست صحيحة في سياق الجملة

a stellar and versatile acress whose combination of . . .

و الصحيح هي actress

لذا سنقوم باستخدام تكنيك bigram و ذلك للاعتماد علي الكلمة السابقة لها لمعرفة مدي احتمال وجود الكلمة المطلوبة

ونري هنا كيف اننا نقوم اولا بتناول كلمة مقترحة و ليكن actress

ثم نقوم بحساب نسبة مجيئها عقب الكلمة السابقة , ثم نسبة مجئ الكلمة التالية لها عقب هذه الكلمة

فيتم حساب الكلمة السابقة ثم الحالية ثم التالية معا عبر ضربهم معا , و يتم تكرار هذا مع الكلمة المقترحة الثانية , ونقارن بينهم

"a stellar and versatile **acress** whose combination of sass and glamour…"

Counts from the Corpus of Contemporary American English with add-1 smoothing

P(actress|versatile)=.000021 P(whose|actress) = .0010

P(across|versatile) =.000021 P(whose|across) = .000006

P("versatile actress whose") = .000021*.0010 = 210 x10^{-10}

P("versatile across whose") = .000021*.000006 = 1 x10^{-10}

-

ماذا عن النوع الثاني من الكلمات , وهي ان الكلمة خطأ , لكن هي في ذاتها كلمة صحيحة في اللغة الانجليزية

نري هنا عدد من الجمل فيها كلمات هي صحيحة و لكن في موضع خطأ هكذا , و هناك تقديرات ان من 25 الي 40 % من الكلمات الخطأ من هذا النوع

…leaving in about fifteen ***minuets*** to go to her house.
The design ***an*** construction of the system…
Can they ***lave*** him my messages?
The study was conducted mainly ***be*** John Black.

25-40% of spelling errors are real words Kukich 1992

و تكمن المشكلة اننا لا نعرف هل هناك خطأ أم لا , واذا كان هناك خطأ فنحن لا نعرف في اي كلمة

و كأن هذا النوع من الأخطاء قد يتواجد في اي جملة , دون ان نعلم هل هو موجود ام لا

و يبدأ الحل عبر تناول كل كلمة علي حدة , ثم نمسك هذه الكلمة و نستخلص منها ترشيحات عديدة قريبة منها , ثم نقوم باختيار خوارزم معين لقياس الاحتمالية

فلو كان لدينا جملة فيها عدد من الكلمات , نقوم بعمل كلمات مناظرة للأولي , ثم كلمات مناظرة للثانية و هكذا :

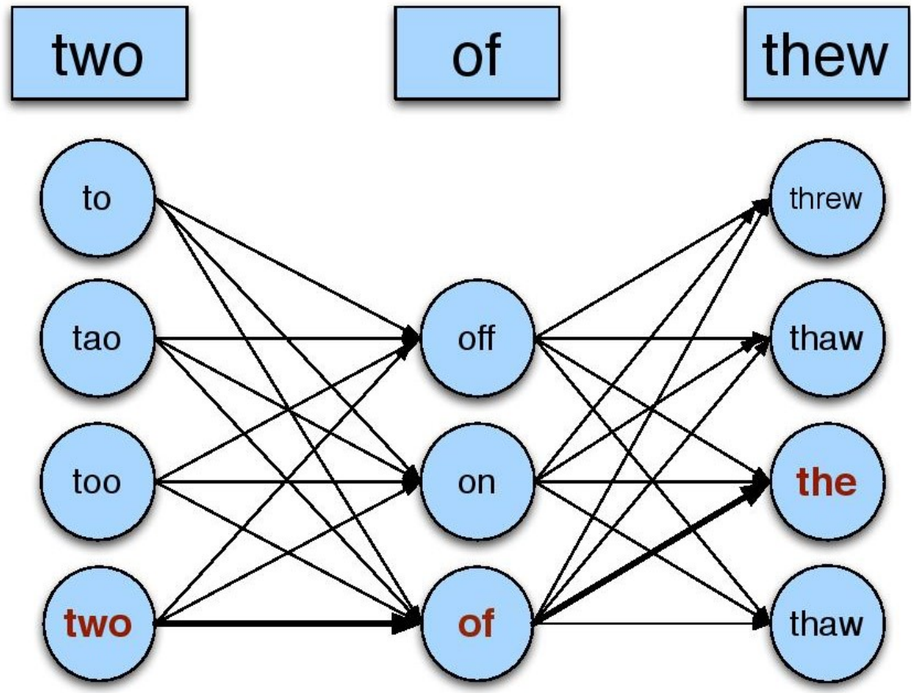
Given a sentence $w_1, w_2, w_3, \ldots, w_n$

Generate a set of candidates for each word w_i

- Candidate(w_1) = {w_1, w'_1 , w''_1 , w'''_1 ,…}
- Candidate(w_2) = {w_2, w'_2 , w''_2 , w'''_2 ,…}
- Candidate(w_n) = {w_n, w'_n , w''_n , w'''_n ,…}

Choose the sequence W that maximizes P(W)

ثم القيام بعمل شبكة من جميع الكلمات المناظرة معا , باسلوب ال combination , ثم نقوم بحساب احتمالية كل مسار علي حدة , ثم اختيار المسار ذو الاحتمالية الاكبر



و يتم هذا عبر احد خوارزميات اللغة

مع الاهتمام باستخدام اسلوب probability of no error , وهي معناها ما احتمالية ان تكون الكلمة صحيحة و ليست خاطئة , اي ان كلمة of المكتوبة هي بالفعل المطلوبة و ليست كلمة اخري

و يتم حساب هذه القيمة عبر التدريب , حينما يتم تصحيحها , فلو كان هناك في مقالة معينة تم تصحيح كلمة of الي كلمة اخري مرة واحدة في حين انها تواجدت بشكل صحيح 10 مرات , تكون النسبة 0.9 , و هكذا

What is the channel probability for a correctly typed word?

P("the"|"the")

Obviously this depends on the application

- .90 (1 error in 10 words)
- .95 (1 error in 20 words)
- .99 (1 error in 100 words)
- .995 (1 error in 200 words)

و بالتالي حينما نقوم بحساب اي كلمة هي المناسبة عبر 5 اختيارات , نقوم اولا بحساب نسبة أن تكون الكلمة صحيحة (0.95) و حساب ان تكون خاطئة كما هو موضح

و لكن الاهم حساب عبر unigram نسبة تواجد هذه الكلمات , فنري ان كلمة the وهي غير مطابقة للاصل و قيمتها فقط 0.000007 , ولكن نسبة تواجدها (عبر ال unigram) هي 0.02

لذا فيجب ان يتم ضرب القيمتين , ونري وقتها ان كلمة the هي التي سيتم اختيارها , لانها منتشرة اكثر بكثير

و نقابل مثل هذه الأمور حينما نكتب كلمة نادرة , وهي قريبة من كلمة منتشرة , مثل اسم مدينة ain او lyon الفرنسية و التي تقترب من كلمة air , lion فغالبا ما يتم تصحيح الكلمة تلقائيا

x	w	x|w	P(x|w)	P(w)	10^9 P(x|w)P(w)
thew	the	ew|e	0.000007	0.02	144
thew	thew		0.95	0.00000009	90
thew	thaw	e|a	0.001	0.0000007	0.7
thew	threw	h|hr	0.000008	0.000004	0.03
thew	thwe	ew|we	0.000003	0.00000004	0.0001

_

و هنا عدد من النصائح في التعامل مع تصحيح الكلمات

- اذا كنت شديد الثقة من تصحيح الكلمة : فقم بعمل autocorrect
- اذا كنت بثقة اقل : فقم بعمل اقتراح افضل كلمة
- اذا كنت بثقة اقل : قم بعمل قائمة من الاقتراحات
- اذا كنت بثقة اقل : قم بعمل علامة علي الكلمة

غالبا ما يقوم نظام ال noisy ليس بضرب الاحتمالات فقط , ولكن يرفع الاحتمال الثاني للأس lamda

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(x \mid w) P(w)^{\lambda}$$

علي اعتبار ان يكون هذا احد المعاملات المستخدمة في التحكم في الموديل بحيث قد يزيد او يقل حسب الداتا

كما ان هناك قواعد منتشرة في التصحيح مثل :

- احذف الحروف المتكررة (باستثناء الحروف التي يشتهر تكرارها , مثل cc , ee)
- احذف الحروف التي لا تتكرر معا (gn) الا لو كانت capital
- استبدل الحروف التي يكون فيها خطأ بنسبة منتشرة ph to f .. ant to ent

هنا عوامل اخري يعتمد عليها تصحيح الحروف , منها ايضا اقتراب حروف لوحة المفاتيح من بعضها البعض (كلمة كلمو)

Factors that could influence p(misspelling|word)

- The source letter
- The target letter
- Surrounding letters
- The position in the word
- Nearby keys on the keyboard
- Homology on the keyboard
- Pronunciations
- Likely morpheme transformations

و يمكن استخدام التصنيف لمعرفة الكلمات السابقة و التالية , فلو جاءت كلمتي weather , whether

فنري هل هناك كلمة cloudy في القرب ,او هل هناك verb معين و هكذا

_