N-Grams and Smoothing

course based on Jurafsky and Martin [2009, Chap.4]

UPPSALA
UNIVERSITET
Marie Dubremetz
marie.dubremetz@lingfil.uu.se

Uppsala, April 2014

Presentation Plan

What is an N-gram ?

Definition of N-gram

(from N the mathematical symbol for natural number and $\gamma\rho\alpha\mu\mu\alpha$: the character, the sequence of letters or the word) Contiguous sequence of items from a given text of speech. The items can be letters, words, tag, tokens...

Why are N-grams Important in NLP ?

The observations of Shannon

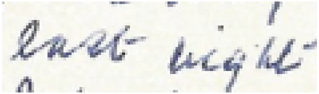
https://www.youtube.com/watch?t=22&v=WyAtOqfCiBw

Why are N-grams Important in NLP ?

Well, can you read the word below ?

Why are N-grams Important in NLP ?

And now, can you read the word below ?



Why are N-grams Important in NLP ?

And now, can you read the word below ?

This was an example of OCR recognition task. But we can imagine other applications :

- Automatic speech recognition
- Spelling correction
- Part-of-speech tagging
- Machine translation
- etc.

Why are N-grams Important in NLP ?

Word prediction

- There are many sources of knowledge that can be used to inform this task, including arbitrary world knowledge.
- But it turns out that you can do pretty well by simply looking at the preceding words and keeping track of some fairly simple counts.
- N-grams counts are easier to implement in a computer than world knowledge or grammar knowledge.

N-grams Used for Word Prediction.

Language modeling

- During this course we will deal with N-grams of words, thus with word prediction
- We can model the word prediction task as the prediction of the conditional probability of a word given previous words in the sequence : $P(w_n|w1, w2...w_n - 1)$
- We'll call a statistical model that can do this a **Language Model**

N-grams Used for Word Prediction.

Definitions of your task(s)

- Task 1 : create a language model. You have corpus, and you need to output the prediction rules based on this corpus.
- Task 2 : apply/use this model in a given application.

=> In this course we will teach you how to create a language model (Task 1).

N-grams Used for Word Prediction.

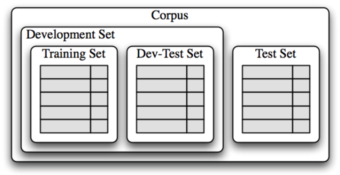
Definitions of your task(s)

- Task 1 : create a language model. You have corpus, and you need to output the prediction rules based on this corpus.
- Task 2 : apply/use this model in a given application.

=> In this course we will teach you how to create a language model (Task 1).

Definition

The corpus on which you calculate the prediction rules is called a **training corpus**.



2 ways of making a language model

- Based on the raw frequencies of the corpus without any smoothing : MLE (maximum likelihood estimate)
- Smoothed versions (Laplace, Good-Turing, Interpolation, Back-off)

Maximum Likelihood Estimate

Formula for the MLE of Unigrams

The unsmoothed maximum likelihood estimate of the unigram probability of the word w_i is its count c_i normalized by the total number of word tokens N :

$$P(w_i) = \frac{c_i}{N}$$

Exercise on Maximum Likelihood Estimate

Formula for the MLE of unigrams

$$P(w_i) = \frac{c_i}{N}$$

Corpus

<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>

According to MLE, find the value of P(I) :

1. $\frac{1}{20}$
2. $\frac{3}{20}$
3. 3

Maximum Likelihood Estimate

Formula for the MLE of bigrams

$$P(w_i|w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

Exercise on Maximum Likelihood Estimate

Formula for the MLE of bigrams

$$P(w_i|w_{i-1}) = \frac{count(w_{i-1},w_i)}{count(w_{i-1})}$$

Corpus

$<$s$>$ I am Sam $<$/s$>$

$<$s$>$ Sam I am $<$/s$>$

$<$s$>$ I do not like green eggs and ham $<$/s$>$

According to MLE, find the value of P(I|$<$s$>$) :

1. $\frac{2}{3}$
2. $\frac{3}{2}$
3. 1
4. 2
5. 3

Problem of Maximum Likelihood Estimate

Maximum likelihood estimate does not take into account the sparsity of the training data.

Problem of Maximum Likelihood Estimate

Maximum likelihood estimate does not take into account the sparsity of the training data.

Do you see the problem coming ? 

Smoothing Justification

If we take the metaphora of fishing, as your fish net can contain only sardines and tuna, this does not mean that there are no sharks in the sea...

It is not because you never see a word in your training corpus that this word would never appear ...and vice versa !

Problem

How do we model these statements mathematically ? By the techniques called **smoothing**.

Laplace Smoothing

Reminder : formula of MLE for unigrams

The unsmoothed maximum likelihood estimate of the unigram probability of the word w_i is its count c_i normalized by the total number of word tokens N :

$$P(w_i) = \frac{c_i}{N}$$

Formula of Laplace for unigrams

$$P_{Laplace}(w_i) = \frac{c_i+1}{N+V}$$

Where V is the size of the vocabulary. Write this formula down for the quiz ! (Do not forget what each letters means.)

Laplace Smoothing

Reminder : formula of MLE for unigrams

The unsmoothed maximum likelihood estimate of the unigram probability of the word w_i is its count c_i normalized by the total number of word tokens N :

$$P(w_i) = \frac{c_i}{N}$$

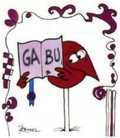
Formula of Laplace for unigrams

$$P_{Laplace}(w_i) = \frac{c_i + 1}{N + V}$$

Where V is the size of the vocabulary. Write this formula down for the quiz ! (Do not forget what each letters means.)

QUIZ of the Shadoks 

The shadoks have only two words vocabulary Ga and Bu.



One day you find the following corpus :

Ga Ga Ga

QUIZ of the Shadoks 

The shadoks have only two words vocabulary « Ga » and « Bu ». One day you explore the shadoks planet and find the following corpus :

Ga Ga Ga

Question

Given this. Can you associate each number to what it represents in the Laplace formula ?

1. $P_{Laplace}(Ga)$	**a** $\frac{4}{5}$
2. $P_{Laplace}(Bu)$	**b** 2
3. V	**c** $\frac{1}{5}$
4. N	**d** 3
5. c_{Ga}	**e** 3
6. c_{Bu}	**f** 0

Laplace Smoothing

Formula of Laplace for Smoothing for bigrams

$$P_{Laplace}(w_i|w_{i-1}) = \frac{C(w_{i-1},w_i)+1}{C_{w_{i-1}}+V}$$

Write it down again !

Laplace Smoothing

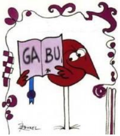
Formula of Laplace for Smoothing for bigrams

$$P_{Laplace}(w_i|w_{i-1}) = \frac{C(w_{i-1},w_i)+1}{C_{w_{i-1}}+V}$$

Write it down again !

QUIZ of the Shadoks

The shadoks have only two words vocabulary Ga and Bu.



One day you find the following corpus :

Bu Bu Bu Ga

Question

What is $P_{Laplace}(Ga|Bu)$?

- $\frac{1}{5}$
- $\frac{2}{5}$
- $\frac{3}{5}$

Good-Turing Smoothing

Intuition used by many smoothing algorithms

- Good-Turing
- Kneser-Ney
- Witten-Bell

Use the count of things we've seen once to help estimate the count of things we've never seen

Good-Turing Smoothing

Imagine you are fishing There are 8 species : carp, perch, pike, trout, salmon, eel, catfish, bass

- You have caught
  - 10 carp, 3 perch, 2 pike, 1 trout, 1 salmon, 1 eel $= 18$ fish
- How likely is it that the next catch is a new species ?
  - $3/18$
- Assuming so, how likely is it that next species is trout ?
  - Must be less than $1/18$

Good-Turing Smoothing

Notation : N_x is the frequency-of-frequency-x $N_{10} = 1$ (carp) $N_1 = 3$ (trout, salmon, eel)

- To estimate total number of unseen species
  - Use number of species (words) we've seen once
  - $P(unseen) = N_1/N = 3/18$
- All other counts are adjusted (down) to give probabilities for unseen

Good-Turing Smoothing Example

	unseen (bass or catfish)	trout
c	0	1
MLE p	$p = \frac{0}{18} = 0$	$\frac{1}{18}$
c^*		$c^*(\text{trout}) = 2 \times \frac{N_2}{N_1} = 2 \times \frac{1}{3} = .67$
GT p^*_{GT}	$p^*_{\text{GT}}(\text{unseen}) = \frac{N_1}{N} = \frac{3}{18} = .17$	$p^*_{\text{GT}}(\text{trout}) = \frac{.67}{18} = \frac{1}{27} = .037$

Complications

- In practice, assume large counts (c $>$ k) are reliable :

$$c^* = c \ \text{ for } c > k$$

- That complicates c*, making it :

$$c^* = \frac{(c+1)\frac{N_{c+1}}{N_c} - c\frac{(k+1)N_{k+1}}{N_1}}{1 - \frac{(k+1)N_{k+1}}{N_1}}, \ \text{ for } 1 \leq c \leq k.$$

- Also : we assume singleton counts c $=$ 1 are unreliable, so treat N-grams with count of 1 as if they were c $=$ 0
- Also : need the N_k to be non-zero, so we need to smooth (interpolate) N_K counts before computing c* from them

Other Information to Combine

- Another really useful source of knowledge
  - If we are estimating trigram P(z|xy)
  - But count(xyz) is zero
  - Use info from bigram p(z|y)
  - Or even unigram p(z)
- How to combine this trigram, bigram, unigram info in a valid fashion ?

Backoff vs. Interpolation

- Backoff : use trigram if you have it, otherwise bigram, otherwise unigram
- Interpolation : mix all three

Summary

- **N-grams** of words are used to create **language models**
- Those language models are based on probabilities.
- The probabilities are learnt on corpora.
- Applying MLE is a good start but not sufficient if we want to construct a good set of probabilities.
- Smoothing techniques exist. We have presented a sample of them (Laplace and Good-Turing).

References

Daniel Jurafsky and James H Martin. *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, volume 163 of *Prentice Hall Series in Artificial Intelligence*. Prentice Hall, 2009.

A good pedagogical material on Turing smoothing : http://kochanski.org/gpk/teaching/0401Oxford/GoodTuring.pdf

Legal informations, copyrights etc.