On an Empirical Study of Smoothing Techniques for a Tiny Language Model

F. Mezzoudj[1,2], M. Loukam [2], A. Benyettou[1]
[1] USTO-MB, Oran, Algeria, [2] UHBC, Chlef, Algeria

IPAC, Batna, November 24^{th}, 2015

Introduction

- A **language model** (LM) is a probabilistic model that assigns probabilities to any sequence of words.
- The LM is an important module in differents systems such as Automatic speech recognition (ASR), Machine translation (MT), OCR & Handwriting recognition, etc.
- We compare the behavior of many smoothing algorithms that have been developed in speech and NLP fields, using a text corpus extracted from French radio show transcription.

Language Models

- The Language modelling is the task of learning a language model that assigns hight probabilities to well formed sentences and plays a crutial role in differents systems, such as ;
- **Example** :
  - **ASR** : p(i saw a van) $>>$ p(eyes awe of an ..)
  - **MT** : une personne intelligente :
    p(a smart person) $>>$ p(a person smart)

Language Models

A sentence $s = w_1 w_2 w_3 ... w_l$;

$$p(s) = p(w_1 w_2 w_3 ... w_l)$$

according to conditional probabilities ..

$$= p(w_1)p(w_2|w_1)p(w_3|w_1 w_2)..p(w_l|w_1..w_{l-1})$$

$$= p(w_1)\prod_{i=2}^{l} p(w_i|w_1..w_{l-1})$$

according to Markov Assumption ..

$$\approx p(w_1)\prod_{i=n}^{l} p(w_i|w_{i-n+1}..w_{i-1})$$

Markov Assumption

An **n-gram** is sequence of n words !

Example : sentence = **<s> is a person <s/>**

- Unigrams : $p(w_1 w_2 ... w_n) \approx \prod_i p(w_i)$
  - **[<s>],[is], [a], [person],[<s/>].**
- Bigrams : $p(w_1 w_2 ... w_n) \approx \prod_i p(w_i | w_{i-1})$
  - **[<s>, is],[is, a], [a, person], [person, <s/>].**
- Trigrams : $p(w_1 w_2 ... w_n) \approx \prod_i p(w_i | w_{i-2} w_{i-1})$
  - **[<s>, is, a],[is, a, person], [a, person, <s/>].**

Perplexity

- The perplexity is the measure of the LM complexity ; and it is the geometric meaning of the word branching factor.
    - For a text T (used for a test),

$$PP(T) = 2^{-\frac{1}{W_T} log_2 p(T)} \tag{1}$$

W_T is the length of text T (by words).

- On comparing the perplexities of 2 LMs, the lesser is for the better LM.

Smoothing methods

- The n-gram LM probabilities of an n-gram model that has not be seen before (in train) can be zero.
- A smoothing is adjusting low probabilities such as zero probabilities upward, and high probabilities downward !
- Sometimes it helps to use less context for contexts we haven't learned much about !
    - **Backoff** : use trigram if you have good evidence, otherwise bigram, otherwise unigram
    - **Interpolation** : mix unigram, bigram, trigram

Smoothing methods

- The ***Additive smoothing*** is one of the simplest backoff methods.
- To avoid zero probabilities, we pretend that each n-gram occurs δ times more than it actually does, where $0 < \delta \leq 1$, and we use :

$$p_{add}(w_i|w_{i-n+1}^{i-1}) = \frac{\delta + c(w_{i-n+1}^{i})}{\delta|V| + \sum_{w_i} c(w_{i-n+1}^{i})}$$

where V is the vocabulary considered.

Smoothing methods

- The ***Ney's Absolute discount*** with standard interpolated version, the n-gram probability is interpolated with lower-order estimates, the equation we use :

$$p_{abs}(w_i|w_{i-n+1}^{i-1}) = \frac{max\{c(w_{i-n+1}^{i}) - D, 0\}}{\sum_{w_i} c(w_{i-n+1}^{i})}) \\ + (1 - \lambda_{w_{i-n+1}^{i-1}})p_{abs}(w_i|w_{i-n+2}^{i-1}) \quad (2)$$

where D is estimated by : $D = \frac{n_1}{n_1+2n_2}$, and n_1, n_2 are the total number of n-grams with one and two counts, in the training data.

Experiments/Results

Table : Corpus size statistics

Corpus	Sentence count	Word count
Etape-train	18 083	285 735
Etape-held-out	1 004	12 790
Etape-test	1 004	18 427

- A pretreatment process was applied on all data.
- All our LMs are built with the SRILM toolkit.

Experiments/Results

In order to achieve a good comparison, we have focused on finding the best n for the n-gram according to our context.

Table : Perplexity for different values of n (for n-gram)

n (n-gram)	ppl	ppl1
1	493.5	910.6
2	189.8	318.7
3	**174.9**	244.5
4	174.9	244.7
5	175.1	244.9

Experiments/Results

Table : Perplexity of different smoothing algorithms on test data

Method	ppl
Standard (Good-Turing)	174.9
Good-turing optimized	177.2
Add-Smooth	195.9
Absolute-discount-backoff	175.9
Absolute-discount-Interpolation	179.9
Original K-Ney-backoff	169.7
Original K-Ney-interpolation	167.5
Modified K-Ney-backoff	173.5
Modified K-Ney-interpolation	**165.4**

Conclusions & perspectives

- The Smoothing is a fundamental technique for statistical modelling language.
- The interpolated models are best then backoff models : the Modified Kneser-Ney using interpolation achieves better result.
- we have measured the performance of algorithms through the PP to check the generalisation ability of the LMs.
- For future work, we will use LM ; with the appropriate smoothing algorithm, as a module for specific application ; speech recognition.

Thank you

for your attention !