Language Models

Marcello Federico
FBK-irst Trento, Italy

MT Marathon 2010
Outline

- Role of LM in ASR and SMT
- N-gram Language Models
- Evaluation of Language Models
- Frequency Smoothing
- Frequency Discounting
- Is smoothing necessary?
- ARPA LM representation
- New IRSTLM Toolkit
Fundamental Equation of ASR

Goal: find the words $w^*$ in a speech signal $x$ such that:

$$w^* = \arg \max_w \Pr(x \mid w) \Pr(w)$$

Problems:

- language modeling (LM): estimating $\Pr(w)$
- acoustic modeling (AM): estimating $\Pr(x \mid w)$
- search problem: computing (1)

AM sums over hidden state sequences $s$ a Markov process of $(x, s)$ from $w$

$$\Pr(x \mid w) = \sum_s \Pr(x, s \mid w)$$

Hidden Markov Model: hidden states "link" speech frames to words.
Fundamental Equation of SMT

**Goal:** find the English string $f$ translating the foreign text $f$ such that:

$$e^* = \arg\max_e \Pr(f \mid e) \Pr(e)$$  \hspace{1cm} (2)

**Problems:**

- **language modeling** (LM): estimating $\Pr(e)$
- **translation modeling** (TM): estimating $\Pr(f \mid e)$
- **search** problem: computing (2)

TM sums over hidden alignments $a$ a stochastic process generating $(f, a)$ from $e$.

$$\Pr(f \mid e) = \sum_a \Pr(f, a \mid e)$$

**Alignment Models:** hidden alignments ”link” foreign words with English words.
• Parallel data are samples of observations \((x, w)\) and \((f, e)\)

• AM and TM can be machine-learned without observing \(s\) and \(a\).
Log-linear phrase-based SMT

- Translation hypotheses are ranked by:

\[ e^* = \arg\max_e \sum_a \sum_k \lambda_k \log h_k(e, f, a) \]

- Phrases are finite string (cf. n-grams)

- Hidden variable \( a \) embeds:
  - segmentation of \( f \) and \( e \) into phrases
  - alignment of phrases of \( f \) with phrases of \( e \)

- Feature functions \( h_k() \) include:
  - Translation Model: appropriateness of phrase-pairs
  - Distortion Model: word re-ordering
  - Language Model: fluency of target string
  - Length Model: number of target words

- Role of the LM is exactly the same as for the classical approach:
  - to score translations generated incrementally by the search algorithm!
N-gram Language Model

Goal: given a text $w^T_1 = w_1 \ldots, w_t, \ldots, w_T$ we can compute its probability by:

$$\text{Pr}(w^T_1) = \text{Pr}(w_1) \prod_{t=2}^{T} \text{Pr}(w_t | h_t)$$ (3)

where $h_t = w_1, \ldots, w_{t-1}$ indicates the history of word $w_t$.

- $\text{Pr}(w_t | h_t)$ becomes difficult to estimate as the history $h_t$ grows.
- hence, we take the $n$-gram approximation $h_t \approx w_{t-n+1} \ldots w_{t-1}$

  e.g. Full history: $\text{Pr}($Parliament$ | I$ declare resumed the session of the European$)$

  3-gram: $\text{Pr}($Parliament$ |$ the European$)$

The choice of $n$ determines the complexity of the LM (# of parameters):

- bad: no magic recipe about the optimal order $n$ for a given task
- good: language models can be evaluated quite cheaply
Language Model Evaluation

- Indirect: impact on task (e.g. BLEU score for MT)
- Direct: capability of predicting words

The perplexity (PP) measure is defined as:

\[ PP = 2^{LP} \text{ where } LP = -\frac{1}{M} \log_2 p(w_1^M) \]  \hspace{1cm} (4)

- \( w_1^M \) is a sufficiently long test sample and \( p(w_1^M) \) is the LM probability

Properties:
- \( 0 \leq PP \leq |V| \) (size of the vocabulary \( V \))
- predictions are as good as guessing among \( PP \) equally likely options

Good: there is typical strong correlation between PP and BLUE scores!

\[ ^1 \text{[Exercise 1. Find PP of 1-gram LM } p(T) = 0.5 \text{ on } T \ H \ T \ H \ T \ H \ T \ T \ H \ T \ T \ H \text{.} ] \]
Estimating $n$-gram probabilities is not trivial due to:

- **parameter space**: with 10,000-word $V$ we can form one trillion 3-grams!
- **data sparseness**: most of 3-grams are rare events even in large corpora.

**Relative frequency estimate**: MLE of any discrete conditional distribution is:

$$f(w \mid x y) = \frac{c(x y w)}{\sum_w c(x y w)}$$

where counts $c(\cdot)$ are taken over a large training corpus.

**Problem**: relative frequencies in general over-fit the training data

- if the test sample contains a ”new” $n$-gram $PP \to +\infty$
- with 4-grams or 5-grams LM this is largely the most frequent case!

**We need frequency smoothing!**
Frequency Smoothing

**Issue:** $f(w \mid x y) > 0$ only if $w$ was observed after $x y$ in the training data.

**Idea:** for each $w$ take off some fraction of probability from $f(w \mid x y)$ and redistribute the total to words never observed after $x y$.

- the discounted frequency $f^*(w \mid x y)$ satisfies:

  $$0 \leq f^*(w \mid x y) \leq f(w \mid x y) \quad \forall x, y, w \in V$$

  Notice: in general $f^*(w \mid x y)$ does not sum up to 1!

- the ”total discount” is called zero-frequency probability $\lambda(x y)^{2}$:

  $$\lambda(x y) = 1.0 - \sum_{w \in V} f^*(w \mid x y)$$

**How to redistribute the total discount?**

\[\text{Notice: by convention } \lambda(x y) = 1 \text{ if } f(w \mid x y) = 0 \text{ for all } w, \text{ i.e. } c(x y) = 0.\]
Discounting Example

Frequency Smoothing

Total frequency amount for never observed words. It is zero for the unsmoothed F.

N-GRAMS "AIMING AT ↔"
Frequency Smoothing

**Insight:** redistribute $\lambda(x \ y)$ according to the lower-order smoothed frequency.

Two major hierarchical schemes to compute the smoothed frequency $p(w \mid x \ y)$:

- **Back-off**, i.e. select the best available $n$-gram approximation:

  $p(w \mid x \ y) = \begin{cases} f^*(w \mid x \ y) & \text{if } f^*(w \mid x \ y) > 0 \\ \alpha_{xy} \lambda(x \ y)p(w \mid y) & \text{otherwise} \end{cases}$ \hspace{1cm} (5)

  where $\alpha_{xy}$ is an appropriate normalization term.\(^3\)

- **Interpolation**, i.e. sum up the two approximations:

  $p(w \mid x \ y) = f^*(w \mid x \ y) + \lambda(x \ y)p(w \mid y).$ \hspace{1cm} (6)

Smoothed frequencies are learned bottom-up, starting from 1-grams ...

\(^3\)[Exercise 2. Find and expression for $\alpha_{xy}$ s.t. $\sum_w p(w \mid x \ y) = 1.$]
Unigram smoothing permits to treat out-of-vocabulary (OOV) words in the LM.

Assumptions:

- $|U|$ is an upper-bound estimate of the size of language vocabulary
- $f^*(w)$ is strictly positive on the observed vocabulary $V$
- $\lambda$ is the total discount reserved to OOV words

Then: 1-gram back-off and interpolation collapse to:

$$p(w) = \begin{cases} 
  f^*(w) & \text{if } w \in V \\
  \lambda(|U| - |V|)^{-1} & \text{otherwise}
\end{cases}$$

(7)

Notice: LMs make also other approximations when an OOV word $x$ appears:

$$p(w \mid h_1 x h_1) = p(w \mid h_2) \quad \text{and} \quad p(x \mid h) = p(x)$$

Important: use a common value $|U|$ when comparing/combining different LMs!
Discounting Methods

Witten-Bell estimate (WB) [Witten and Bell, 1991]

- **Insight:** learn $\lambda(x\ y)$ by counting ”new word” events in 3-grams $x\ y\ *$
  - corpus: $x\ y\ u\ x\ x\ y\ t\ t\ x\ y\ u\ w\ x\ y\ w\ x\ y\ t\ u\ x\ y\ u\ x\ y\ t$
  - then $\lambda(x\ y) \propto$ number of ”new word” events (i.e. 3)
  - and $f^*(w \mid x\ y) \propto$ relative frequency (linear discounting)

- **Solution:**

$$\lambda(x\ y) = \frac{n(x\ y\ *)}{c(x\ y) + n(x\ y\ *)} \quad \text{and} \quad f^*(w \mid xy) = \frac{c(x\ y\ w)}{c(x\ y) + n(x\ y\ *)}$$

where $c(x\ y) = \sum_w c(x\ y\ w)$ and $n(x\ y\ *) = |\{w : c(x\ y\ w) > 0\}|$.  

- **Pros:** easy to compute, robust for small corpora, works with artificial data.
- **Cons:** underestimates probability of frequent $n$-grams

---

4[Exercise 3. Compute $f^*(u \mid x\ y)$ with WB on the above artificial text.]
Discounting Methods

- interpolation and back-off with WB discounting
- trigram LMs estimated on the English Europarl corpus
- logprobs of 3-grams of type \textit{aiming at \_} observed in training

![Graph showing logprobs of 3-grams 'aiming at \_' in Europarl]

- peaks correspond to very probable 2-grams interpolated with $f^*$ respectively: at that, at national, at European
- Practically, \textit{interpolation and back-off perform similarly}
Discounting Methods

Absolute Discounting (AD) [Ney and Essen, 1991]

- **Insight:**
  - discount by subtracting a small constant $\beta \ (0 < \beta \leq 1)$ from each counts
  - estimate $\beta$ by maximizing the leaving-one-out likelihood of the training data

- **Solution:** (notice: one distinct $\beta$ for each n-gram order)

$$
 f^* (w \mid x y) = \max \left\{ \frac{c(xyw) - \beta}{c(xy)}, 0 \right\}
$$

which gives

$$
\lambda(xy) = \beta \frac{\sum_{w : c(xyw) > 1} 1}{c(xy)}
$$

where $\beta \approx \frac{n_1}{n_1 + 2n_2} < 1$ and $n_r = |\{x y w : c(x y w) = r\}|$.

- **Pros:** easy to compute, accurate estimate of frequent n-grams.
- **Cons:** problematic with small and artificial samples.

---

5[Exercise 4. Given the text in WB slide find the number of 3-grams, $n_1$, $n_2$, $\beta$, $f^*(w \mid x y)$ and $\lambda(x y)$]

M. Federico, FBK-irst 
M. Federico, FBK-irst
Discounting Methods

**Kneser-Ney method (KN)** [Kneser and Ney, 1995]

- **Insight:** 2-grams counts should be correspond to the "back-off" cases
  - count all "back-off" events in 3-grams of type \(* \ y \ w\) (cf. WB method)
  - corpus: `x y w x t y w t x y w u y w t y w t y w u x y w u u y w`
  - corrected counts \(n(* \ y \ w)\) = number of observed back-offs (i.e. 3)

- **Solution:** (for 3-gram normal counts)

\[
 f^*(w \mid y) = \max \left\{ \frac{n(* \ y \ w) - \beta}{n(* \ y \ *)}, 0 \right\}
\]

which gives

\[
 \lambda(y) = \beta \frac{\sum_{w:n(* \ y \ w) > 1} 1}{n(* \ y \ *)}
\]

where \(n(* \ y \ w)\) = \(|\{x : c(x \ y \ w) > 0\}|\) and \(n(* \ y \ *)\) = \(|\{x \ w : c(x \ y \ w) > 0\}|\)

- **Pros:** better back-off probabilities, can be applied to other methods
- **Cons:** LM cannot be used to compute lower order \(n\)-gram probs
Modified Kneser-Ney (MKN) [Chen and Goodman, 1999]

- **Insight**: specific discounting coefficients for unfrequent $n$-grams
- **Solution**:
  \[
  f^*(w | x y) = \frac{c(x y w) - \beta(c(x y w))}{c(x y)}
  \]
  where $\beta(0) = 0$, $\beta(1) = D_1$, $\beta(2) = D_2$, $\beta(c) = D_{3+}$ if $c \geq 3$, coefficients are computed from $n_r$ statistics, corrected counts used for lower order $n$-grams

- **Pros**: see previous + more fine grained smoothing
- **Cons**: see previous + more sensitiveness to noise

**Important**: LM interpolation with MKN is the most popular training method. Under proper training conditions it gives the best PP and BLEU scores!
Discounting Methods

- interpolation with WB and MKN discounting trained on Europarl
- the plot shows the logprob of observed 3-grams of type aiming at _

![Graph showing logprob of observed 3-grams]

- notice that for less frequent 3-grams WB assigns higher probability
- we have three very high peaks corresponding large corrected counts:
  \( n(*\text{at that})=665 \) \( n(* \text{ at national})=598 \) \( n(* \text{ at European})=1118 \)
- also an interesting peak at rank #26: \( n(* \text{ at very})=61 \)
Discounting Methods

- train: interpolation with WB and MKN discounting on Europarl
- test: 3-grams of type \texttt{aiming at \_} are from the Google 1TWeb sample

The trend is the same but MKN outperforms WB smoothing.
If you don’t believe, check the next slide ....
Discounting Methods

- **train**: interpolation with WB and MKN discounting on Europarl
- **test**: 3-grams of type *aiming at _* are from the Google 1TWeb sample
- **plot**: cumulative score differences between MKN and WB on top 1000 3-grams

![Cumulative score difference MSB-WB](image-url)
Is LM Smoothing Necessary?

- **LM Quantization** [Federico and Bertoldi, 2006]
  - **Idea:** one codebook for each n-gram/back-off level
  - **Pros:** improves storage efficiency
  - **Cons:** reduces discriminatory power
  - Experiments with 8bit quantization on ZH-EN NIST task showed:
    * 2.7% BLUE drop with a 5-gram LM trained on 100M-words
    * 1.6% BLUE drop with a 5-gram LM trained on 1.7G words.

- **Stupid back-off** [Brants et al., 2007]
  - simple smoothing, no correct normalization

\[
p(w \mid x y) = \begin{cases} 
  f(w \mid x y) & \text{if } f(w \mid x y) > 0 \\
  k \cdot p(w \mid y) & \text{otherwise}
\end{cases}
\]

where \( k = 0.4 \) and \( p(w) = c(w)/N \).
Is LM Smoothing Necessary?

Stupid back-off (SB) versus Modified Kneser-Ney (KN)

From [Brants et al., 2007].

- Conclusion: proper smoothing useful up to 1 billion word training data?
ARPA File Format (srilm, irstlm)

Represents both interpolated and back-off n-gram LMs

- format: \( \log(\text{smoothed-freq}) :: \text{n-gram} :: \log(\text{back-off weight}) \)
- computation: look first for smoothed-freq, otherwise back-off

\[
\begin{align*}
\text{\textbackslash data}\text{\textbackslash} \\
\text{ngram 1=} & 86700 \\
\text{ngram 2=} & 1948935 \\
\text{ngram 3=} & 2070512 \\
\text{1-grams:} & \\
& -2.88382 \quad ! \quad -2.38764 \\
& -2.94351 \quad \text{world} \quad -0.514311 \\
& -6.09691 \quad \text{dublin} \quad -0.15553 \\
& \ldots \\
\text{2-grams:} & \\
& -3.91009 \quad \text{world} \quad -0.351469 \\
& -3.91257 \quad \text{hello world} \quad -0.24 \\
& -3.87582 \quad \text{hello dublin} \quad -0.0312 \\
& \ldots \\
\text{3-grams:} & \\
& -0.00108858 \quad \text{hello world} \quad ! \\
& -0.000271867 \quad , \quad \text{hi hello} \quad ! \\
& \ldots \\
\text{end}\text{\textbackslash} \\
\end{align*}
\]

\[
\begin{align*}
\log \text{Pr}(!| \text{hello dublin}) &= -0.0312 + \log \text{Pr}(!| \text{dublin}) \\
\log \text{Pr}(!| \text{dublin}) &= -0.15553 - 2.88382
\end{align*}
\]
Main Features [Federico et al., 2008]

- Single thread training for standard LMs
  - all major smoothing methods: WB, AD, MKN, ...
  - LM pruning, internal/external interpolation, adaptation

- Distributed training for huge LMs
  - simple smoothing methods: interpolation with WB, KN
  - split dictionary into balanced \( n \)-gram prefix lists
  - collect \( n \)-grams for each prefix lists
  - estimate and merge single LMs for each prefix list

- Space optimization
  - \( n \)-gram collection uses dynamic storage to encode counters
  - distributed LM estimation just requires reading disk files
  - probs and back-off weights are quantized
  - run-time LM data structure is loaded on demand

- LM caching
  - computations of probs, access to internal lists, LM states, ....
References


natural language modelling. In Proceedings of the IEEE International Conference on Acoustics,