INF4820

Crash Course on Probability Theory + N-Gram Modeling

Erik Velldal

University of Oslo

Sep. 22, 2009

Statistical Methods in NLP

Every time I fire a linguist, system performance goes up.
(Fred Jelinek, IBM, 1980s)

- Quoted in a zillion papers introducing NLP, and in every party speech at the conference banquettes.
- Related to an important debate in the field: The division of symbolic and statistical approaches, and the merging of the two.

Statistical Methods in NLP

Every time I fire a linguist, system performance goes up.
(Fred Jelinek, IBM, 1980s)

- Quoted in a zillion papers introducing NLP, and in every party speech at the conference banquettes.
- Related to an important debate in the field: The division of symbolic and statistical approaches, and the merging of the two.

- Today we'll be looking at one of the core modeling techniques of statistical / probabilistic NLP: n-gram models.
  - Models language use as sequences with associated probabilities.

N-Gram Models

What are they good for?

- Sequence Analysis
- Depending on the application, n-grams can be defined on different levels, e.g. on the character level, word level, phoneme level...

N-Gram Models

What are they good for?

- Sequence Analysis
- Depending on the application, n-grams can be defined on different levels, e.g. on the character level, word level, phoneme level...
- Example applications in NLP:
  - Speech Recognition
  - Handwriting Recognition
  - OCR
  - Spelling Correction
  - Spelling Assistance (e.g. auto-completion)
  - MT

N-Gram Models

What are they good for?

- Sequence Analysis
- Depending on the application, n-grams can be defined on different levels, e.g. on the character level, word level, phoneme level...
- Example applications in NLP:
  - Speech Recognition
  - Handwriting Recognition
  - OCR
  - Spelling Correction
  - Spelling Assistance (e.g. auto-completion)
  - MT
- Also widely used in other fields (e.g. text compression, gene sequence analysis, DNA classification...)

Crash Course on Probability Theory

First we introduce some terminology

- The sample space is a set Ω of elementary outcomes.
  - For example, let Ω represent the outcome of throwing a die: $\Omega = \{one, two, three, four, five, six\}$

Crash Course on Probability Theory

First we introduce some terminology

- The sample space is a set Ω of elementary outcomes.
  - For example, let Ω represent the outcome of throwing a die: $\Omega = \{one, two, three, four, five, six\}$
- Events $(A, B, C, \ldots)$ are subsets of this set, such as $\{one\}$, $\{two, four, six\}$, $\{three, six\}$, etc.

Crash Course on Probability Theory

First we introduce some terminology

- The sample space is a set Ω of elementary outcomes.
  - For example, let Ω represent the outcome of throwing a die: $\Omega = \{one, two, three, four, five, six\}$
- Events $(A, B, C, \ldots)$ are subsets of this set, such as $\{one\}$, $\{two, four, six\}$, $\{three, six\}$, etc.
- A probability measure P is a function from events to the interval $[0, 1]$.
- $P(A)$ is the probability of event A.

Some Basic Rules

The three axioms of probability theory...

- $P(A) \geq 0$ for all events A (non-negativity)
- $P(\Omega) = 1$ (unit measure)
- $A \cap B = \emptyset \quad \Rightarrow \quad P(A \cup B) = P(A) + P(B)$ (additivity for disjoint events)

Some Basic Rules

The three axioms of probability theory...

- $P(A) \geq 0$ for all events A (non-negativity)
- $P(\Omega) = 1$ (unit measure)
- $A \cap B = \emptyset \quad \Rightarrow \quad P(A \cup B) = P(A) + P(B)$ (additivity for disjoint events)

And some of their consequences

- $P(\bar{A}) = 1 - P(A)$
- $P(\emptyset) = 0$
- $P(B \backslash A) = P(B) - P(A \cap B)$
- If $A \subseteq B$ then $P(A) \leq P(B)$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ (the addition rule)

Conditional Probability and Independence

- The notion of conditional probability lets us capture the influence of partial knowledge about the outcome.
- The conditional probability of A given B is defined as

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- The probability that we will observe A given that we have already observed B. The fraction of B's probability mass that also covers A.
- $P(A)$ is often referred to as the *a priori* or *prior* probability, while $P(A|B)$ is referred to as the *a posteriori* or *posterior* probability.

Conditional Probability and Independence (cont'd)

The Multiplication Rule

- The numerator in our equation for conditional probability can itself be "conditionalized" using the multiplication rule:
- $P(A \cap B) = P(A)P(B|A) = P(B)P(A|B)$

Conditional Probability and Independence (cont'd)

The Multiplication Rule

- The numerator in our equation for conditional probability can itself be "conditionalized" using the multiplication rule:
- $P(A \cap B) = P(A)P(B|A) = P(B)P(A|B)$

The Chain Rule

- Generalizes the multiplication rule to multiple events:
- $P(A \cap B \cap C \cap D \cap \ldots) = P(A)P(B|A)P(C|A \cap B)P(D|A \cap B \cap C) \ldots$

Conditional Probability and Independence (cont'd)

The Multiplication Rule

- The numerator in our equation for conditional probability can itself be "conditionalized" using the multiplication rule:
- $P(A \cap B) = P(A)P(B|A) = P(B)P(A|B)$

The Chain Rule

- Generalizes the multiplication rule to multiple events:
- $P(A \cap B \cap C \cap D \cap \ldots) = P(A)P(B|A)P(C|A \cap B)P(D|A \cap B \cap C) \ldots$

Independence

- Two events A and B are independent if $P(A \cap B) = P(A)P(B)$.
- In other words, $P(A|B) = P(A)$.

Bayes' Rule

- Lets us swap the order of dependence between events. If we know one we can get to the other.
- Bayes' rules states that

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes' Rule

- Lets us swap the order of dependence between events. If we know one we can get to the other.
- Bayes' rules states that

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Often we are just interested in which event $\tilde{A}$ out some set that is most likely given B. Since the denominator is just a normalizing constant, we can skip it:

$$\tilde{A} = \arg\max_A P(A|B) = \arg\max_A P(B|A)P(A)$$

Random Variables

- Instead of dealing with with the sample space directly (which will be different for each application), random variables provides an extra layer of abstraction that lets us deal with events in a more uniform way.
- A discrete random variable X is a function from the sample space Ω into a finite set of numerical values $\{x_1, x_2, \ldots, x_n\}$.

Random Variables

- Instead of dealing with with the sample space directly (which will be different for each application), random variables provides an extra layer of abstraction that lets us deal with events in a more uniform way.
- A discrete random variable X is a function from the sample space Ω into a finite set of numerical values $\{x_1, x_2, \ldots, x_n\}$.
- The probability mass function (pmf) p for a random variable X gives us the probability of a given numerical value:

$$p(x) = p(X = x) = P\left(\{\omega \in \Omega : X(\omega) = x\}\right)$$

Random Variables

- Instead of dealing with with the sample space directly (which will be different for each application), random variables provides an extra layer of abstraction that lets us deal with events in a more uniform way.
- A discrete random variable X is a function from the sample space Ω into a finite set of numerical values $\{x_1, x_2, \ldots, x_n\}$.
- The probability mass function (pmf) p for a random variable X gives us the probability of a given numerical value:

$$p(x) = p(X = x) = P\left(\{\omega \in \Omega : X(\omega) = x\}\right)$$

- For a discrete random variable X, let x_i be the value corresponding to the event A_{x_i} in Ω. We then have that:

$$\sum_{x_i \in X} p(X = x_i) = \sum_{A_{x_i} \in \Omega} P(A_{x_i}) = P(\Omega) = 1$$

Random Variables (cont'd)

- Note that, for our purposes, we can often collapse the distinction between values in Ω and X, so we will occasionally be writing (strictly speaking nonsensical) things like $p(x_i = \text{'banana'})$, instead of "$p(X = x_i)$ *where* $x_i \in X$, $\omega \subseteq \Omega$, $X(\omega) = x_i$, *and* $\omega = \{\text{'banana'}\}$".

Random Variables (cont'd)

- Note that, for our purposes, we can often collapse the distinction between values in Ω and X, so we will occasionally be writing (strictly speaking nonsensical) things like $p(x_i = \text{'banana'})$, instead of "$p(X = x_i)$ *where* $x_i \in X$, $\omega \subseteq \Omega$, $X(\omega) = x_i$, *and* $\omega = \{\text{'banana'}\}$".
- We often want to define several random variables over the same sample space, resulting in a joint probability distribution, e.g. $p(x, y) = p(X = x, Y = y)$.
  - A joint pmf can be seen as analogous to a probability function for intersection of events.
  - For example, we can define the conditional pmf as $p(x|y) = \frac{p(x,y)}{p(y)}$, and use the chain rule such as $p(x, y, z) = p(x)p(y|x)p(z|x, y)$.

Random Variables (cont'd)

- Note that, for our purposes, we can often collapse the distinction between values in Ω and X, so we will occasionally be writing (strictly speaking nonsensical) things like $p(x_i = \text{'banana'})$, instead of "$p(X = x_i)$ *where* $x_i \in X$, $\omega \subseteq \Omega$, $X(\omega) = x_i$, *and* $\omega = \{\text{'banana'}\}$".
- We often want to define several random variables over the same sample space, resulting in a joint probability distribution, e.g. $p(x, y) = p(X = x, Y = y)$.
  - A joint pmf can be seen as analogous to a probability function for intersection of events.
  - For example, we can define the conditional pmf as $p(x|y) = \frac{p(x,y)}{p(y)}$, and use the chain rule such as $p(x, y, z) = p(x)p(y|x)p(z|x, y)$.
- A sequence of random variables $X_1, X_2, X_3, \ldots$ over the same sample space Ω, is also known as a random or stochastic process.

But we were supposed to be talking about n-grams

- When using n-gram models for language modeling, the starting point is to view language as a stochastic process.

But we were supposed to be talking about n-grams

- When using n-gram models for language modeling, the starting point is to view language as a stochastic process.

The plan for the rest of this lecture:

- The structure of n-gram models
- Estimation
- Evaluation
- The problem of sparse data

From strings to words, and back again

- By the chain rule of joint probabilities, the probability of a sequence of words can be factorized as:

$$p(w_1, \ldots, w_k) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \ldots p(w_k|w_1, \ldots, w_{k-1})$$

From strings to words, and back again

- By the chain rule of joint probabilities, the probability of a sequence of words can be factorized as:

$$p(w_1, \ldots, w_k) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \ldots p(w_k|w_1, \ldots, w_{k-1})$$

- However, modeling this full distribution would require too many parameters...

From strings to words, and back again

- By the chain rule of joint probabilities, the probability of a sequence of words can be factorized as:

$$p(w_1, \ldots, w_k) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \ldots p(w_k|w_1, \ldots, w_{k-1})$$

- However, modeling this full distribution would require too many parameters...
- We simplify the estimation problem by the so-called *Markov assumption*: The probability of a given word is taken to only depend on the $n-1$ words preceding it.
- The probability of a string $w_1, \ldots, w_k$ is just the product of its individual word probabilities, computed as:

$$p_n(w_1^k) = \prod_{i=1}^{k} p(w_i|w_{i-n+1}^{i-1})$$

From strings to words, and back again

- By the chain rule of joint probabilities, the probability of a sequence of words can be factorized as:

$$p(w_1, \ldots, w_k) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2)\ldots p(w_k|w_1, \ldots, w_{k-1})$$

- However, modeling this full distribution would require too many parameters...
- We simplify the estimation problem by the so-called *Markov assumption*: The probability of a given word is taken to only depend on the $n-1$ words preceding it.
- The probability of a string $w_1, \ldots, w_k$ is just the product of its individual word probabilities, computed as:

$$p_n(w_1^k) = \prod_{i=1}^{k} p(w_i|w_{i-n+1}^{i-1})$$

Special markers for sentence boundaries: `<s>` and `</s>`.

Just to be clear about terminology...

- An n-gram in itself is just a sub-sequence of n elements from a sequence $w_1, \ldots, w_k$.
- In an n-gram model the probability of the ith word in a given sequence is conditioned on the last $n-1$ words in its history...
- so we can have *bigram* models ($n=2$), *trigram* models ($n=3$), *fourgram* models ($n=4$), etc.
- More generally, models aimed at computing the probability of sequences of words are known as language models (LMs).

Estimation

- How to get our hands on the probabilities?

Estimation

- How to get our hands on the probabilities? By counting:

$$P(\text{bananas}|\text{i}, \text{like}) = \frac{C(\text{i}, \text{like}, \text{bananas})}{C(\text{i}, \text{like}, *)}$$

- The probabilities are given by the relative frequencies of the observed outcomes, i.e. as they occur in our training corpus.

Estimation

- How to get our hands on the probabilities? By counting:

$$P(\text{bananas}|\text{i}, \text{like}) = \frac{C(\text{i}, \text{like}, \text{bananas})}{C(\text{i}, \text{like}, *)}$$

- The probabilities are given by the relative frequencies of the observed outcomes, i.e. as they occur in our training corpus.
- = Maximum likelihood estimation / MLE

Some Considerations When Counting Words

- What is to count as a word depends on the application.
- Tokenization
- Normalization of case, spelling variants, clitics, abbreviations, numerical expressions, punctuation...
- Lemmatization. Base forms vs full word forms.

Evaluation

- Extrinsic Evaluation
  - AKA application-based or end-to-end evaluation
  - Measure the effect on performance within the embedding application (e.g. the MT-system or the speech-recognizer that the LM is a part of).

Evaluation

- Extrinsic Evaluation
  - AKA application-based or end-to-end evaluation
  - Measure the effect on performance within the embedding application (e.g. the MT-system or the speech-recognizer that the LM is a part of).
- Intrinsic
  - Quick and cheap evaluation based on the probability that the model assigns to unseen test data.
  - Typically based on the perplexity measure. The perplexity of a model p with respect to the test data W_1^N is $p(w_1, \ldots, w_N)^{-\frac{1}{N}}$.
  - The perplexity is inversely related to the average probability assigned to the words in the test sample, so minimizing the perplexity is equivalent to maximizing the probability of the test set.

Training Data vs Test Data

- We use the training set to *estimate* our models and use the test set to *evaluate* them.

Training Data vs Test Data

- We use the training set to *estimate* our models and use the test set to *evaluate* them.
- Testing on the training data would give unrealistically high expectations about performance in a real application.
- Even with a perfect score when testing on the training data, we would probably see a drastic fall in performance on unseen data.

Training Data vs Test Data

- We use the training set to *estimate* our models and use the test set to *evaluate* them.
- Testing on the training data would give unrealistically high expectations about performance in a real application.
- Even with a perfect score when testing on the training data, we would probably see a drastic fall in performance on unseen data.
- Related: Overfitting. Why is the MLE a poor estimator?

Problems With Our Maximum Likelihood Estimates

- Data sparseness
  - Regardless of corpus size, perfectly acceptable phrases will be missing.
  - The creativity of language.

Problems With Our Maximum Likelihood Estimates

- Data sparseness
  - Regardless of corpus size, perfectly acceptable phrases will be missing.
  - The creativity of language.
- Unknown words
- Zero counts (mixes badly with the factorization into word probabilities)

Problems With Our Maximum Likelihood Estimates

- Data sparseness
  - Regardless of corpus size, perfectly acceptable phrases will be missing.
  - The creativity of language.
- Unknown words
- Zero counts (mixes badly with the factorization into word probabilities)

Some remedies

- Make provisions for out-of-vocabulary words (OOVs).
  - Open vs closed vocabulary

Problems With Our Maximum Likelihood Estimates

- Data sparseness
  - Regardless of corpus size, perfectly acceptable phrases will be missing.
  - The creativity of language.
- Unknown words
- Zero counts (mixes badly with the factorization into word probabilities)

Some remedies

- Make provisions for out-of-vocabulary words (OOVs).
  - Open vs closed vocabulary
- Make sure all n-grams receive a non-zero count (smoothing).

Smoothing

- AKA discounting
- General idea: take some of the probability mass of frequent events, and redistribute it to less frequent or unseen events.
- Makes the distribution less "spiked".
- Simplest approach: Add-One smoothing.

Smoothing

- AKA discounting
- General idea: take some of the probability mass of frequent events, and redistribute it to less frequent or unseen events.
- Makes the distribution less "spiked".
- Simplest approach: Add-One smoothing.
- Other LM techniques aimed at overcoming problems with unseen events:
  - Back-off and interpolated models
  - Class-based models