Natural Language Processing
Lecture 2

Lecture by Dr. Kfir Bar
Typeset by Steven Karas

2019-03-12
Last edited 20:59:25 2019-03-12

Disclaimer These notes are based on the lectures for the course Natural Language Processing, taught by Dr. Kfir Bar at IDC Herzliyah in the spring semester of 2018/2019. Sections may be based on the lecture slides prepared by Dr. Kfir Bar.

1 Homework

The first homework project will be published tonight.

2 Probability Theory Review

A sample space is the set of all possible outcomes of an experiment.

An event is a subset of the sample space.

The rest of the probability section is given on slides 5-27.

3 Language Model

There are two general approaches to language modeling:

Formal grammars give a hard model that strictly defines membership or not. A probabilistic model of a language gives a probability that a string is a member of a language. The probabilistic approach is more useful.

Models have many applications, usually to refine results from other applications. To refine the results of speech recognition: "I ate a cherry" is more likely than "Eye eight uh Jerry".

3.1 Word sequence probabilities

Given a word sequence as a stochastic process:

$$w_1^n = w_1 \dots w_n$$

where states are words. The probability distribution is:

$$\Pr[w_1^n] = \Pr[w_1] \Pr[w_2|w_1] \Pr[w_3|w_1^2] \dots \Pr[w_n|w_1^{n-1}] = \prod_{i=1}^{n} \Pr[w_k|w_1^{k-1}]$$

The Markov assumption is that future behavior of a system only depends on the recent history. For a kth-order Markovian model, the next state only depends on the last k states.

3.2 Terminology

Corpus a body of text used to train models. Typically a portion of the corpus is set aside for later

Types distinct words in a corpus. Sometimes called the vocabulary

Tokens surface words

3.3 N-gram models

An N-gram model uses only $N-1$ words of prior context. In these models, we use special marker tokens for tokens outside the body of content (both start and end).

The probability approximation is:

$$\Pr\left[w_1^n\right] = \prod_{k=1}^{n} \Pr\left[w_k \mid w_{k-N+1}^{k-1}\right]$$

The conditional probabilities can be estimated based on the relative frequency of observed sequences ($C(w_k^n)$ counts the number of occurrences of the word sequence w_k^n):

$$\Pr\left[w_n \mid w_{n-N+1}^{n-1}\right] = \frac{C\left(w_{n-N+1}^{n-1}\, w_n\right)}{C\left(w_{n-N+1}^{n-1}\right)}$$

It happens that relative frequencies are a maximum likelihood estimator as they maximize the probability of the observed sequences T given the model parameters θ.

$$\hat{\theta} = \arg\max_{\theta} \Pr[T \mid \theta]$$

Example: Shakespeare A 1-gram model trained on Shakespeare gives nonsense. A 2-gram model is a bit awkward, but at least gives something that looks like text. A 3-gram model reads comfortably for phrases, but the sentences make no sense. A 4-gram model reads very much like Shakespeare.

The example is available via Google Colab.

3.4 Evaluating models

We use two main measures: perplexity and entropy.

3.4.1 Entropy

Measures the uncertainty in a model. Given a random variable X with probabilities $\{p_1, \ldots, p_n\}$ then we say that entropy is maximized if $p_1 = \ldots = p_n = \frac{1}{n}$. Defined as the expected negative log probability:

$$H(X) = -\sum_{x \in X} \Pr[x] \log_2 \Pr[x]$$

3.4.2 Cross Entropy

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} \Pr[x,y] \log \Pr[x,y]$$

The cross entropy of a true distribution p and a model distribution m is defined as:

$$H(p,m) = -\sum_{x} p[x] \log m[x]$$

Note that $H(p)$ is a lower bound for $H(p,m)$

Language model cross entropy Events are sequences of words:

$$-\sum_{x} p[w_1, \ldots, w_n] \log m[w_1, \ldots, w_n]$$

But we want the per-word rate, so normalize:

$$-\frac{1}{n} \sum_{x} p[w_1, \ldots, w_n] \log m[w_1, \ldots, w_n]$$

But we really care about sequences of words:

$$
\begin{aligned}
H(p,m)_{w_1^n} &= \lim_{n\to\infty} -\frac{1}{n}\sum_i p[w_1^n]\log m[w_1^n] \\
&= -\frac{1}{N}\log m[w_1^n]
\end{aligned}
$$

3.4.3 Conditional Entropy

$$H(H \mid Y) = H(X,Y) - H(Y)$$

3.4.4 Mutual information

Measures the mutual dependence between variables:

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

3.5 Perplexity

The weighted average number of choices a random variable has to make:

$$\mathit{perplexity}(X) = 2^{H(X)}$$

Better models of the unknown probability will have lower perplexity. They are less surprised by the test data.

4 Next week

Next week there will not be a lecture.

References

[1] Yoav Goldberg. A primer on neural network models for natural language processing. *CoRR*, abs/1510.00726, 2015.

[2] Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., 2009.

[3] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.