Cross Entropy for Measuring quality in models

Peter Nabende
Alfa Informatica, CLCG, RuG
p.nabende@rug.nl

Entropy

- Entropy finds its origins from information theory and is used to quantify information
- Intuitively, entropy quantifies the uncertainty involved when encountering a random variable X
- The random variable X ranges over whatever we are predicting (e.g. named entities, characters, etc)
- and has a particular probability function, call it $p(x)$

Entropy

- Suppose we have a set of *events* whose probabilities of occurrence are $p(x_1), p(x_2), ..., p(x_n)$
- We would like to measure how much uncertainty is associated with the *events*
- Such a measure, say $H(x)$ (Shannon, 1948) should have the following properties

  1) $H(x)$ should be continuous in the $p(x_i)$

  2) If all the $p(x_i)$ are equal, $p(x_i) = \frac{1}{n}$, then H should be a monotonic increasing function of $H(x)$

  3) If a choice is broken down into two successive choices, the original $H(x)$ should be the weighted sum of the individual values of $H(x)$

Entropy

- The only $H(x)$ satisfying the three above assumptions is of the form:

$$H(x) = -K\sum_{i=1}^{n} p(x_i)\log p(x_i)$$

where K is a positive constant. $H(x)$ is referred to as Entropy of the probability distribution over the events

- The choice of the logarithmic base will correspond to the choice of the unit for measuring information
- For a sequence of observations $S = \{s_1, s_2, ..., s_n\}$, we will be computing the entropy of the sequence

Entropy

- For a sequence of characters in a string, we will be interested in the entropy of observing the characters in the string

$$H(s_1, s_2, ..., s_n) = -\sum_{S_1^n \in A} p(S_1^n) \log p(S_1^n)$$

where A is an alphabet of characters

Cross Entropy (Jurafsky and Martin, 2009)

- Cross entropy is used to compare probability distributions
- It allows us to use some model *m*, which is a model of *p* (i.e., an approximation to *p*)
- The cross entropy of two probability distributions *p* and *m* for a random variable X is written as:

$$H(p, m) = -\sum_i p(x_i) \log(m(x_i))$$

- It should be noted that cross entropy is not a symmetric function

$$H(p, m) \neq H(m, p)$$

Cross Entropy (Jurafsky and Martin, 2009)

- The cross entropy $H(p, m)$ is an upper bound on the true entropy $H(p)$. For any model m:

$$H(p) \leq H(p, m)$$

- If $p = m$, the cross entropy is said to be at a minimum and

$$H(p, m) = H(p) .$$

- The closer the cross entropy $H(p, m)$ is to the true entropy $H(p)$, the more accurate the model m (or the better m is an approximation of p)
- Cross entropy can therefore be used to compare approximate models
  - Between two models m_1 and m_2, the more accurate model will be the one with the lower cross entropy

Cross Entropy: Example

- The table below shows the actual probability distribution of a random variable X and two approximate distributions m_1 and m_2.

X	x_1	x_2	x_3	x_4	x_5
p	0.3	0.2	0.1	0.2	0.2
m_1	0.2	0.2	0.2	0.2	0.2
m_2	0.3	0.1	0.1	0.1	0.4

Table 1: Probability distribution of a random variable with two approximations

- The entropy of X is $H(p) = -\sum_i p(x_i) \log(p(x_i)) = 0.672$

Cross Entropy: Example

- The cross entropy for m_1 is:

$$H(p, m_1) = -\sum_i p(x_i) \log(m_1(x_i)) = 0.699$$

- The cross entropy for m_2 is:

$$H(p, m_2) = -\sum_i p(x_i) \log(m_2(x_i)) = 0.736$$

- In this example, the uniform distribution m_1 is better than m_2 at approximating the true distribution p

- The cross entropy becomes much more useful when we do not know the actual (true) probability distribution .

The case of a Language Grammar (Jurafsky and Martin, 2009)

- For a language grammar, we can be interested in the entropy of some sequence of words $W = \{w_0, w_1, w_2, ..., w_n\}$
- The entropy of a random variable that ranges over all finite sequences of words of length n in some language L can be computed as:

$$H(w_0, w_1, w_2, ..., w_n) = -\sum_{W_1^n \in L} p(W_1^n) \log p\left(W_1^n\right)$$

- For a number of words, we can have the entropy rate (per word entropy) given as:

$$\frac{1}{n} H(W_1^n) = -\frac{1}{n} \sum_{W_1^n \in L} p(W_1^n) \log p\left(W_1^n\right)$$

- To measure the true entropy of a language we need to consider sequences of infinite length

The case of a Language Grammar (Jurafsky and Martin, 2009)

- If a language is thought of as a stochastic process L that produces a sequence of words, its entropy rate $H(L)$ is defined as:

$$H(L) = -\lim_{n\to\infty} \frac{1}{n} H(w_1, w_2, ..., w_n)$$

$$= -\lim_{n\to\infty} \frac{1}{n} p(w_1, w_2, ..., w_n) \log p(w_1, w_2, ..., w_n)$$

- If a language is stationary and ergodic, the Shannon-McMillan-Breiman theorem gives us:

$$H(L) = \lim_{n\to\infty} -\frac{1}{n} \log p(w_1, w_2, ..., w_n)$$

- A language is stationary if the probability distribution of the words do not change with time. It is ergodic if its statistical properties can be deduced from a single, sufficiently long sequence of words

The case of a Language Grammar (Jurafsky and Martin, 2009)

- The cross-entropy of m on p will be

$$H(p,m) = \lim_{n\to\infty} -\frac{1}{n}\sum_{W\in L} p(w_1,\ldots,w_n)\log m(w_1,\ldots,w_n)$$

- Note that this assumes that convergence will occur for an existing limit
- Following the Shannon-McMillann-Breiman theorem again, for a stationary ergodic language, the cross entropy of a real probability density function $m(.)$ for estimating the probability distribution of the langauge is written as:

$$H(p,m) = \lim_{n\to\infty} -\frac{1}{n}\log m(w_1,\ldots,w_n)$$

Cross Entropy for evaluating pair HMM Scoring algorithms

- A pair HMM has two observation sequences $(s : t)$ as opposed to one observation sequence in a standard HMM

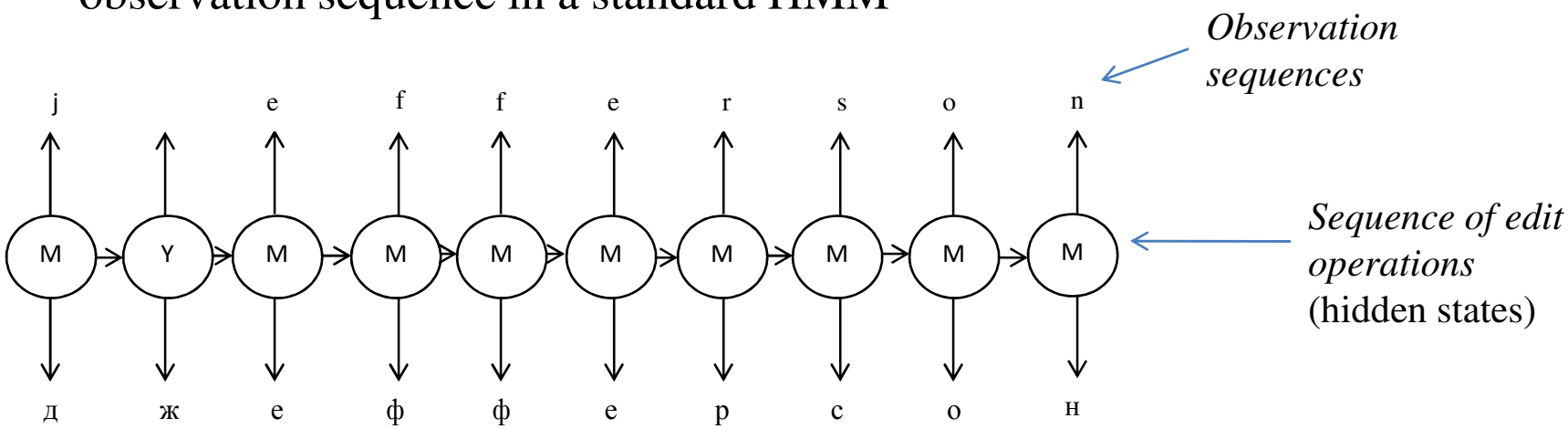

Fig.1: Illustration of alignment for an english name “jefferson” and its Russian transliteration following the pair-HMM concept

- The pair HMM has been used in identification of matching bilingual entity names for languages using different alphabets (English and Russian)

Cross Entropy for evaluating pair HMM Scoring algorithms

- The pair HMM system takes in as input a pair of sequences and outputs a similarity score for the input pair
- For obtaining similarity scores, two algorithms are used: the *Forward* and *Viterbi* algorithm. The task is to identify which algorithm best estimates the similarity between two strings
- The cross entropy for the pair HMM on the probability of observing a pair of sequences is given as:

$$H(p,m) = -\sum_{s\in A_1, t\in A_2} p(s_1:t_1,\ldots,s_T:t_T)\log m(s_1:t_1,\ldots,s_T:t_T)$$

- We draw the pairs of sequences according to the probability distribution p, but sum the log of their probabilities according to m

Cross Entropy for evaluating pair HMM Scoring algorithms

- However, we do not have a target distribution. Instead, we have a corpus that we can exploit in comparing at least two models
- The notion of corpus cross entropy (log probability) is used
  - Given, a corpus C of size n consisting of tokens c_i ,…,c_n , the log probability of a model m on this corpus is defined as:

$$H_C(m) = -(1/n) * \sum_i \log m(c_i))$$

where summation is done over tokens in the corpus

- It can be proven that as n tends to infinity, the corpus cross entropy becomes the cross entropy for the true distribution that generated the corpus

Cross Entropy for evaluating pair HMM Scoring algorithms

- To prove the equivalence of the corpus cross entropy with the true cross entropy, it must be assumed that the corpus has a stationary distribution.
- The proof depends on the fact that the Maximum Likelihood Estimation goes to the true probability distribution as the size of the corpus goes to infinity

- It is not exactly correct to use the result for cross entropy in NLP applications because the above assumption is clearly wrong for languages (Manning and Schutze, 2001)
- Nonetheless, for a given corpus, we can assume that a language is near enough to unchanging. This will be an acceptable approximation to truth (Askari, 2006)

Cross Entropy for evaluating pair HMM Scoring algorithms

- For the case of the pair HMM, the corpus comprises of pairs of entity names
- We consider each pair of names $(s^i : t^i)$ to be a token c_i in the corpus
- Therefore, the log probability of a pair HMM algorithm m on the corpus can be written as:

$$H_C(m) = -(1/n) * \sum_{i=1}^{n} \log(m(s^i : t^i))$$

where $m(s^i : t^i)$ is the estimated probability according to a model m for the pair of names $(s^i : t^i)$

Cross Entropy for evaluating pair HMM Scoring algorithms

- It is also possible to consider character alignments to constitute tokens in the corpus
- In that sense, $(s^i : t^i)$ will be the i^{th} character alignment in the corpus
- Estimating character alignment probabilities can only be possible if the corpus comprises aligned characters, moreover manually corrected

- In either of the cases above (i.e whether character alignments or pairs of entity names are used), it is important to see whether there is any chance of the log probability convergin with increase in corpus size (See figure 2. on the next slide for pair HMM)

Variation of Corpus Cross Entropy with corpus size *n* for two pair HMM scoring algorithms

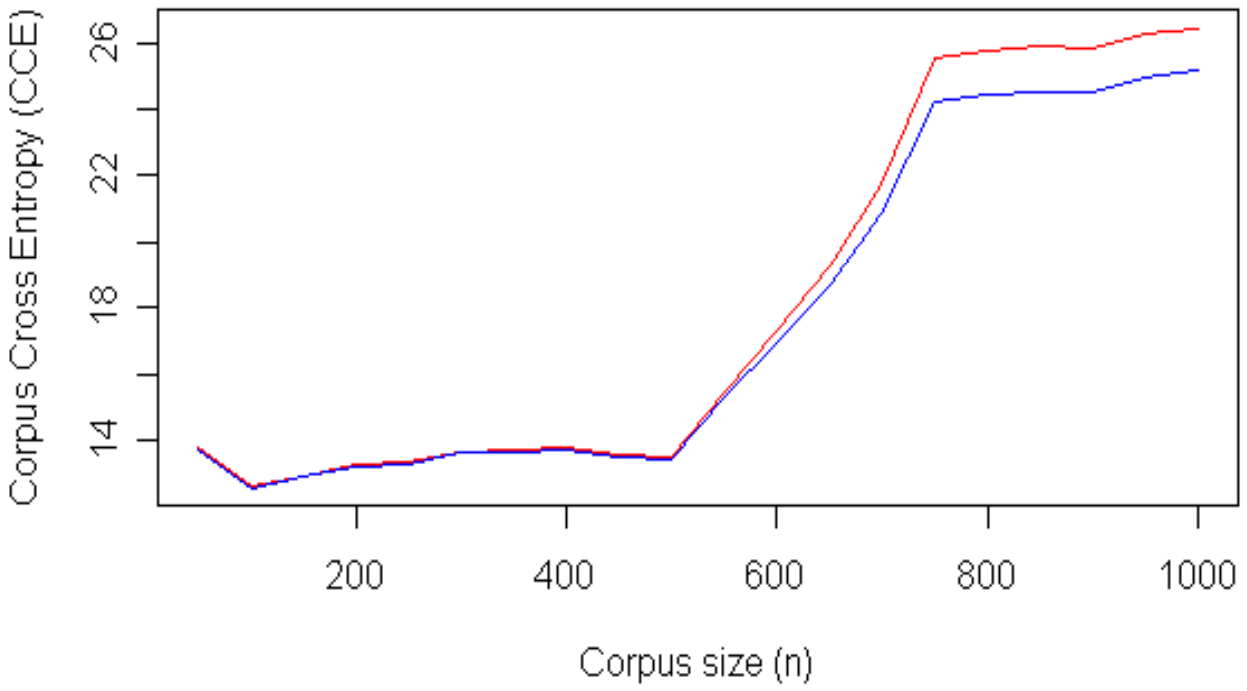

Fig.2: Variation of Corpus Cross Entropy with corpus size *n* (of entity name pairs) for the Forward algorithm (blue) and Viterbi algorithm (red)

Cross Entropy for evaluating pair HMM Scoring algorithms

- Table 1 shows Corpus Cross Entropy (CCE) for two algorithms: Viterbi and Forward on a corpus size of 1000 pairs of English-Russian entity names

Algorithm	Name-pair CCE (for n = 1000
Viterbi	26.4132
Forward	25.1951

Table 1: Cross Entropy results for English-Russian matching entity names

- The Corpus Cross Entropy results suggest that the Forward algorithm is slightly more accurate than the Viterbi algorithm

References

1. Claude E. Shannon. (1948). A Mathematical Theory of Communication., *Bell System Technical Journal*, vol. 27, pp. 379-423, 623-656, July, October, 1948
2. Daniel Jurafsky and James H. Martin. (2009). *Speech and Language Processing*. Pearson Education, Inc., Upper Saddle River, New Jersey 07458 (Section 4.10)
3. James F. Allen. Lecture Notes for Corpus Driven Natural Language Processing: Lecture 6 – Entropy and Information Theory http://www.cs.rochester.edu/james/CSC248/ Retrieved 02nd March 2009.
4. Jean Mark Gawron. Lecture Notes for Statistical Methods in Computation Linguistics: Cross Entropy. http://www-rohan.sdsu.edu/~gawron/stat/crossntropy.htm Accessed 02nd March 2009
5. Chris Manning and Hinrich Schutze. (1999). *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA.
6. Mina Askari. (2006). *Information Theoretic Evaluation of Change Prediction Models for Large-Scale Software*. Masters Thesis in Computer Science, University of Waterloo, Ontario, Canada.