The Answer to a Question at ACL08

Xiaojin Zhu

August 16, 2011

1 An Apparent Paradox

There was an interesting question during the ACL08 presentation, summarized as follows. Let L be a language generated by a bigram language model (LM), e.g.

$$L = \{\langle d \rangle ABB, \langle d \rangle AAB, \ldots\}.$$

Now consider another language L^r which is the same as L except that all documents have their word orders reversed:

$$L^r = \{\langle d \rangle BBA, \langle d \rangle BAA, \ldots\}.$$

For any document in L and its reverse in L^r, the Bag-of-word (BOW) representation is the same. This means that the bigram LM recoverable by our algorithm from BOWs of L and L^r must be the same. It is puzzling since L and L^r seem to have different bigram distributions.

2 No Paradox After All

A short answer is: L^r in general is *not* a language that can be generated by any bigram LM, i.e., there is no "bigram LM of L^r" to talk about.

It is important to note that a necessary condition for exact bigram LM recovery is that the language is indeed generated by some underlying bigram LM. L is such a language by definition. But L^r may not be. To see this, consider the simple example in the paper. The underlying bigram LM that generates L is

$$P(A|\langle d \rangle) = r,\ P(A|A) = p,\ P(B|B) = r.$$

We further assume that all documents have length 4, including $\langle d \rangle$. We can then enumerate all possible documents in L, together with their probability. These are shown as the first and second columns in the table below. By reversing the word orders, we obtain the documents in L^r, as in the third column. Their *observed probability* is the same as in the second column.

Does L^r correspond to some bigram LM? Assume it does, with the parameters

$$P(A|\langle d \rangle) = x,\ P(A|A) = y,\ P(B|B) = z.$$

This leads to the *computed probability* in the fourth column:

L	prob	L^r	computed prob for L^r
$\langle d\rangle AAA$	rp^2	$\langle d\rangle AAA$	xy^2
$\langle d\rangle AAB$	$rp(1-p)$	$\langle d\rangle BAA$	$(1-x)(1-z)y$
$\langle d\rangle ABA$	$r(1-p)(1-q)$	$\langle d\rangle ABA$	$x(1-y)(1-z)$
$\langle d\rangle ABB$	$r(1-p)q$	$\langle d\rangle BBA$	$(1-x)z(1-z)$
$\langle d\rangle BAA$	$(1-r)(1-q)p$	$\langle d\rangle AAB$	$xy(1-y)$
$\langle d\rangle BAB$	$(1-r)(1-q)(1-p)$	$\langle d\rangle BAB$	$(1-x)(1-z)(1-y)$
$\langle d\rangle BBA$	$(1-r)q(1-q)$	$\langle d\rangle ABB$	$x(1-y)z$
$\langle d\rangle BBB$	$(1-r)q^2$	$\langle d\rangle BBB$	$(1-x)z^2$

The second and fourth columns need to match. This leads to a system of equations, and the solution is

$$X = r \quad (1)$$

$$Y = p \quad (2)$$

$$Z = q \quad (3)$$

$$r = \frac{1-q}{2-p-q}. \quad (4)$$

Interpretation: If the original language L does not satisfy (4), then L^r is not generated by any bigram LM. Otherwise, L and L^r are actually the same. In the first case, our algorithm would recover the bigram LM for L; in the second case, our algorithm would recover the correct bigram LM for both L and L^r. Paradox resolved.