# Polytomous logistic regression with Shanghainese topic markers

Antti Arppe, Weifeng Han & John Newman

December 20, 2013

## 1 Preparations

Before carrying out the statistical analyses, we need to invoke the polytomous package to make it available within R, having installed the package earlier. As subsequent preliminary steps, we load in the `shanghainese` data frame, and then take a look at its composition, scrutinizing the first six lines (output length by default for the function `head`) and the overall content of the data frame with the `summary` method:

```
> library(polytomous)
> data(shanghainese)
> head(shanghainese)
  TOPIC_MARKER TOPIC_LENGTH TOPIC_POS FUNCTION COMMENT_TYPE GENRE
1           ne            5       ADJ     CONT         CLAU INTER
2           ne            2       ADJ     EMPH         CLAU  MONO
3           ne            3       ADJ     INTR         CLAU  MONO
4           ne            2       ADV  COUNTER         CLAU  MONO
5           ne            2       ADV  COUNTER         PHRA INTER
6           ne            4       ADV     INTR         PHRA  MONO
> summary(shanghainese)
 TOPIC_MARKER  TOPIC_LENGTH     TOPIC_POS      FUNCTION   COMMENT_TYPE
 ne:100       Min.   : 1.000   NOM   :195   INTR   :197   CLAU :356
 a :100       1st Qu.: 2.000   ADJ   : 25   COUNTER: 31   PHRA : 56
 mo:100       Median : 3.000   ADV   : 63   COND   : 58   TAG  : 50
 zi:100       Mean   : 3.888   CLAUSE:131   CONT   : 47   TSTR : 16
 ma:100       3rd Qu.: 5.000   VERB  : 86   EMPH   :167   TSTRD: 22
              Max.   :10.000
    GENRE
 MONO  :258
 CONV  : 79
 INTER :138
 SCRIPT: 25
```

## 2 Univariate analysis

We start the univariate analysis by creating a cross-tabulation of the occurrences of the INTRODUCTORY FUNCTION (TRUE) vs. its absence (FALSE):

```
> table(shanghainese$FUNCTION=="INTR", shanghainese$TOPIC_MARKER)
       ne  a mo zi ma
  FALSE 82 57 67 33 64
  TRUE  18 43 33 67 36
```

We calculate the topic-marker-wise proportions (as percentages) of the INTRODUCTORY FUNCTION (using the second column of the above cross-tabulation and dividing by the total frequency which is the same for all topic markers, i.e. 100):

```
> round(table(shanghainese$FUNCTION=="INTR", shanghainese$TOPIC_MARKER)[2,]*100/100)
ne  a mo zi ma
18 43 33 67 36
```

Likewise, we can also calculate the feature-wise distribution (as percentages) of the INTRODUCTORY FUNCTION among the five topic markers (dividing the second column values by the overall frequency of the feature in question, i.e. 197):

```
> round(table(shanghainese$FUNCTION=="INTR", shanghainese$TOPIC_MARKER)[2,]*100/197)
ne  a mo zi ma
 9 22 17 34 18
```

Next, we calculate the significance of the distribution with chi-squared test, using the `chisq.test` function which is part of the basic R configuration:

```
> chisq.test(table(shanghainese$FUNCTION=="INTR", shanghainese$TOPIC_MARKER))[c("statistic",
+ "parameter","p.value")]
```

We follow this up first by calculating the two asymmetric Theil's *Uncertainty Coefficients* using the `associations` function from the `polytomous` package, i.e. the reduction of uncertainty concerning the feature (Row) given the topic marker (Column), i.e. `uc.RC`, and the reduction of uncertainty concerning the topic marker (Column) given the feature (Row), i.e. `uc.CR`:

```
> associations(table(shanghainese$FUNCTION=="INTR",
+ shanghainese$TOPIC_MARKER))[c("uc.RC","uc.CR")]
$uc.RC
[1] 0.08233657
$uc.CR
[1] 0.03430202
```

Secondly, we use the `chisq.posthoc` function in the `polytomous` package to assess the significance of the cell-wise divergences from homogeneity (i.e. the differences of the cellwise observed values in relation to the respective expected values) using the *standardized Pearson residuals*:

```
> chisq.posthoc(table(shanghainese$FUNCTION=="INTR",
+ shanghainese$TOPIC_MARKER))$cells$std.pearson.residuals
              ne          a         mo         zi         ma
  FALSE  4.8964842 -0.8237076  1.4643691 -6.3150918  0.7779461
  TRUE  -4.8964842  0.8237076 -1.4643691  6.3150918 -0.7779461
```

The same values can also be extracted from the output of the standard `chisq.test` function:

```
> chisq.test(table(shanghainese$FUNCTION=="INTR", shanghainese$TOPIC_MARKER))$stdres
              ne          a         mo         zi         ma
  FALSE  4.8964842 -0.8237076  1.4643691 -6.3150918  0.7779461
  TRUE  -4.8964842  0.8237076 -1.4643691  6.3150918 -0.7779461
```

We can then conveniently also extract from the output of the `chisq.posthoc` function a simplification of the results of the analysis of cellwise divergences using standardized Pearson residuals, with `'+'` indicating a significant cellwise divergence above the expected value, `'-'` a significant cellwise divergence above the expected value, and `'0'` a cellwise value that does not diverge significantly from the expected value. The threshold value for significance has been explicitly specified here as `std.pearson.residual.min=2` (which is the default value) as an argument to the function:

```
> chisq.posthoc(table(shanghainese$FUNCTION=="INTR", shanghainese$TOPIC_MARKER),
+ std.pearson.residual.min=2)$cells$std.pearson.residuals.sign
      ne a mo zi ma
FALSE  + 0  0  -  0
TRUE   - 0  0  +  0
```

Next, we consider the relationship of all values of the categorical variable FUNCTION and the topic markers, again starting with their cross-tabulation:

```
> table(shanghainese$FUNCTION, shanghainese$TOPIC_MARKER)
          ne  a mo zi ma
  INTR    18 43 33 67 36
  COUNTER 13  4  4  4  6
  COND    15  4 27  1 11
  CONT    24  3  1  1 18
  EMPH    30 46 35 27 29
```

Again, we evaluate the significance of the observed values diverging from expected values (representing a homogeneous distribution) using the `chisq.test` function:

```
> chisq.test(table(shanghainese$FUNCTION, shanghainese$TOPIC_MARKER))$p.value
[1] 5.55568e-21
```

As before, we also calculate the two asymmetric *Uncertainty Coefficients* with the help of the `associations` function:

```
> associations(table(shanghainese$FUNCTION,
+ shanghainese$TOPIC_MARKER))[c("uc.RC","uc.CR")]
$uc.RC
[1] 0.1005342
$uc.CR
[1] 0.08606379
```

This time, we skip the actual values of the standardized Pearson residuals and go straight for the simplified results provided by the `chisq.posthoc` function:

```
> chisq.posthoc(table(shanghainese$FUNCTION,
+ shanghainese$TOPIC_MARKER))$cells$std.pearson.residuals.sign
        ne a mo zi ma
INTR     - 0  0  +  0
COUNTER  + 0  0  0  0
COND     0 -  +  -  0
CONT     + -  -  -  +
EMPH     0 +  0  0  0
```

In principle, we could replicate individually the above steps for each of the categorical variables and topic markers in the dataframe. However, we can in practice use the function `nominal` and in particular its `summary` method to create one dataframe with all the results of the univariate analyses for each value of each categorical variable. The function uses a transformation of the multinomial (categorical) variables in the original dataframe `shanghainese` into a number of equivalent binary/logical (TRUE/FALSE) ones using the `multinomial2logical` function in the `polytomous` package, stored in the dataframe `shanghainese.logical`. In all, the five categorical variables consist of 29 distinct values, which are each renamed with the format `variable.value`, e.g. `FUNCTION.INTR`. The warnings are due to some variable values having low frequencies, which make the chi-squared test unreliable. We also transform the numeric variable TOPIC-LENGTH as a factor for the time being, though we need to remember to revert this transformation later in the multivariate analysis:

```
> shanghainese$TOPIC_LENGTH <- factor(shanghainese$TOPIC_LENGTH)
> shanghainese.logical <- multinomial2logical(shanghainese, outcome="TOPIC_MARKER",
+ variables=c("TOPIC_LENGTH", "TOPIC_POS", "FUNCTION", "COMMENT_TYPE", "GENRE"),
+ variable.value.separator=".")
> shanghainese.univariate <- nominal(TOPIC_MARKER ~ ., data=shanghainese.logical)
```

The resultant dataframe `shanghainese.univariate` is then the basis for the univariate results presented below. Notice that the notation of the association measures has been modified, from e.g. `uc.RC` and `uc.CR` to `uc.12` and `uc.21`, respectively, so that the number code '1' refers to the independent variable (corresponding to Row in the analysis above) and '2' to the dependent variable (corresponding to Column in the analysis above):

```
> print(summary(shanghainese.univariate), max.print=NA)
Univariate analysis of categorical variables:
Dependents (2):  TOPIC_MARKER = a, ma, mo, ne, zi

Independents (1): TOPIC_LENGTH.1, TOPIC_LENGTH.2, TOPIC_LENGTH.3, TOPIC_LENGTH.4, TOPIC_LENGTH.5,
TOPIC_LENGTH.6, TOPIC_LENGTH.7, TOPIC_LENGTH.8, TOPIC_LENGTH.9,
TOPIC_LENGTH.10, TOPIC_POS.NOM, TOPIC_POS.ADJ, TOPIC_POS.ADV, TOPIC_POS.CLAUSE,
TOPIC_POS.VERB, FUNCTION.INTR, FUNCTION.COUNTER, FUNCTION.COND, FUNCTION.CONT,
FUNCTION.EMPH, COMMENT_TYPE.CLAU, COMMENT_TYPE.PHRA, COMMENT_TYPE.TAG,
COMMENT_TYPE.TSTR, COMMENT_TYPE.TSTRD, GENRE.MONO, GENRE.CONV, GENRE.INTER,
GENRE.SCRIPT


                    N     alpha.X2       uc.12        uc.21 a ma mo ne zi
TOPIC_LENGTH.1     34 1.308705e-05   0.1092584   0.01686524 0  0  -  0  +
TOPIC_LENGTH.2    134  2.15492e-06  0.06314691   0.02280593 0  0  -  0  +
TOPIC_LENGTH.3     98    0.7161208 0.004194672  0.001289614 0  0  0  0  0
TOPIC_LENGTH.4     75   0.07111676  0.02127343 0.005587336 0  0  -  0  0
TOPIC_LENGTH.5     47  0.000902802  0.06229419   0.01206437 0  0  +  0  -
TOPIC_LENGTH.6     40    0.3609746  0.01626406 0.002817083 0  0  0  0  0
TOPIC_LENGTH.7     24 0.0002912163    0.101846   0.01218679 0  0  +  0  0
TOPIC_LENGTH.8     32 5.957093e-06   0.1328341   0.01962956 0  -  +  0  -
TOPIC_LENGTH.9     11    0.1424449  0.07816731 0.005134808 0  0  +  0  0
TOPIC_LENGTH.10     5    0.1946695   0.1187869 0.004133279 +  0  0  0  0
TOPIC_POS.NOM     195 4.555367e-10  0.07826371   0.03251987 0  0  -  +  +
TOPIC_POS.ADJ      25  0.001176987  0.07742869 0.009550399 0  0  +  0  0
TOPIC_POS.ADV      63 0.0007870944  0.06773456    0.0159384 0  0  -  +  0
TOPIC_POS.CLAUSE  131 1.042881e-13   0.1134282   0.04053398 0  0  +  -  -
TOPIC_POS.VERB     86   0.03770036  0.02194235   0.00625839 +  0  0  0  0
FUNCTION.INTR     197 5.719865e-11  0.08233657   0.03430202 0  0  0  -  +
FUNCTION.COUNTER   31   0.03342759  0.03871251 0.005590871 0  0  0  +  0
```

```
FUNCTION.COND       58 2.846053e-08   0.1214699   0.02708584 -  0  +  0  -
FUNCTION.CONT       47 3.105367e-11   0.1856047   0.03594561 -  +  -  +  -
FUNCTION.EMPH      167   0.03302451  0.01607947  0.006363851 +  0  0  0  0
COMMENT_TYPE.CLAU  356    0.9358436 0.001354353 0.0005051995 0  0  0  0  0
COMMENT_TYPE.PHRA   56    0.7189567 0.005954955  0.001297511 0  0  0  0  0
COMMENT_TYPE.TAG    50    0.7357589 0.006340007  0.001280589 0  0  0  0  0
COMMENT_TYPE.TSTR   16    0.0540782  0.05254578  0.004623915 +  0  0  0  0
COMMENT_TYPE.TSTRD  22    0.3349709  0.02397384   0.00268801 0  0  0  0  0
GENRE.MONO         258 4.687276e-15   0.1120915    0.0482395 +  0  -  0  0
GENRE.CONV          79 4.485982e-11   0.1319345   0.03576907 -  +  0  -  0
GENRE.INTER        138  1.61198e-12   0.1036923   0.03795662 -  -  +  0  0
GENRE.SCRIPT        25    0.2420524  0.02722677  0.003358271 +  0  0  0  0


Association measures:
  statistic.12 ~ statistic(1|2) ~ statistic(independent|dependent) [ ~ statistic(R|C) ]
  statistic.21 ~ statustic(2|1) ~ statistic(dependent|independent) [ ~ statistic(C|R) ]

Posthoc cellwise chi-squared analysis:  std.pearson.residuals.sign
```

## 2.1 Topic markers and topic-length

As an alternative to treating TOPIC-LENGTH as a categorical variable as above, we could scrutinize instead it as a normal numeric variable, and observe whether its distributions vary among the topic markers using Analysis of Variance. As a preliminary step, we have to remember to change the type of the TOPIC-LENGTH variable from a factor back to numeric.

```
> shanghainese$TOPIC_LENGTH <- as.numeric(as.character(shanghainese$TOPIC_LENGTH))
> summary(aov(TOPIC_LENGTH ~ TOPIC_MARKER, data=shanghainese))
              Df Sum Sq Mean Sq F value Pr(>F)
TOPIC_MARKER   4  457.3  114.32   30.32 <2e-16 ***
Residuals    495 1866.5    3.77
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Indeed, TOPIC-LENGTH is significant in distinguishing the topic-markers, some have inherently longer or shorter topic-lengths than others, and the differences in the distributions are significant (overall). The results are the same if we use the more rigorous *Kruskal-Wallis* test that treats topic-length as an ordinal rather than a continuous interval variable.

```
> kruskal.test(TOPIC_LENGTH ~ TOPIC_MARKER, data=shanghainese)

        Kruskal-Wallis rank sum test

data:  TOPIC_LENGTH by TOPIC_MARKER
Kruskal-Wallis chi-squared = 100.5923, df = 4, p-value < 2.2e-16
```

Comparing the individual topic-markers, we get the following picture:

```
> TukeyHSD(aov(TOPIC_LENGTH ~ TOPIC_MARKER, data=shanghainese))
  Tukey multiple comparisons of means
    95% family-wise confidence level
Fit: aov(formula = TOPIC_LENGTH ~ TOPIC_MARKER, data = shanghainese)

$TOPIC_MARKER
       diff         lwr         upr     p adj
a-ne   0.31 -0.44185298  1.06185298 0.7912139
mo-ne  1.61  0.85814702  2.36185298 0.0000001
zi-ne -1.28 -2.03185298 -0.52814702 0.0000397
ma-ne -0.50 -1.25185298  0.25185298 0.3626948
mo-a   1.30  0.54814702  2.05185298 0.0000283
zi-a  -1.59 -2.34185298 -0.83814702 0.0000001
```

```
ma-a  -0.81 -1.56185298 -0.05814702 0.0274468
zi-mo -2.89 -3.64185298 -2.13814702 0.0000000
ma-mo -2.11 -2.86185298 -1.35814702 0.0000000
ma-zi  0.78  0.02814702  1.53185298 0.0376370
```

The pairwise cases with $P < .05$ `p adj` are the ones of interest. Basically, all the topic markers appear to have different TOPIC-LENGTH distributions: *mo* has the longest topics, *zi* the shortest one, and the three others fall in between, which is also evident from the mean values of TOPIC-LENGTH for the TOPIC-MARKERs:

```
> sapply(levels(shanghainese$TOPIC_MARKER),
+    function(i) mean(shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER==i]))
  ne    a   mo   zi   ma
3.86 4.17 5.47 2.58 3.36
```

Alternatively, we can use the more rigorous *Wilcoxon rank-sum test* (which again assumes ordinal rather than continuous interval data) to see whether the differences between the length-wise adjacent topic-markers are significant. Indeed, even with this test, *mo* is clearly apart from the rest as the longest, and *zi* as the shortest, with the three other topic-markers as a group in the middle (which are not significantly distinguishable from their immediate neighbors in the group):

```
> wilcox.test(shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER=="mo"],
+ shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER=="a"])

        Wilcoxon rank sum test with continuity correction

data:  shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER == "mo"] and shanghainese$TOPIC_LENGTH[shanghainese$TOPI
W = 6752.5, p-value = 1.551e-05
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER=="a"],
+ shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER=="ne"])

        Wilcoxon rank sum test with continuity correction

data:  shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER == "a"] and shanghainese$TOPIC_LENGTH[shanghainese$TOPIC
W = 5367.5, p-value = 0.3611
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER=="ne"],
+ shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER=="ma"])

        Wilcoxon rank sum test with continuity correction

data:  shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER == "ne"] and shanghainese$TOPIC_LENGTH[shanghainese$TOPI
W = 5774.5, p-value = 0.05298
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER=="ma"],
+ shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER=="zi"])

        Wilcoxon rank sum test with continuity correction

data:  shanghainese$TOPIC_LENGTH[shanghainese$TOPIC_MARKER == "ma"] and shanghainese$TOPIC_LENGTH[shanghainese$TOPI
W = 6129, p-value = 0.004361
alternative hypothesis: true location shift is not equal to 0
```

# 3 Bivariate analysis

We start the bivariate analysis by one example case scrutinizing the association of two variable values, TOPIC-LENGTH of 1 and INTRODUCTORY FUNCTION. Now we use the dataframe `shanghainese.logical` with $TRUE/FALSE$ as a basis for our crosstabulation. We can note that these two variable values co-occur 15 times:

```
> table(shanghainese.logical[["TOPIC_LENGTH.1"]], shanghainese.logical[["FUNCTION.INTR"]])
       FALSE TRUE
  FALSE   284  182
  TRUE     19   15
```

Using this crosstabulation we can perform similar individual analyses as at the univariate stage, focusing on the two asymmetric *Uncertainty Coefficients* that we can calculate with the `associations` function. We can now notice that knowing that the FUNCTION is INTRODUCTORY or not (Column) reduces our uncertainty of whether the TOPIC-LENGTH is 1 or not (Row) by only $uc.RC = .001$, whereas knowing that the TOPIC-LENGTH is 1 or not (Row) reduces our uncertainty of whether the FUNCTION is INTRODUCTORY or not (Column) by even less, i.e. $uc.CR = .0005$.

```
> associations(table(shanghainese.logical[["TOPIC_LENGTH.1"]],
+ shanghainese.logical[["FUNCTION.INTR"]]))[c("uc.RC","uc.CR")]
$uc.RC
[1] 0.001353318
$uc.CR
[1] 0.0005014306
```

Again, we could perform the calculations individually on all pairings of each value of each categorical variable considered in our linguistic analysis. In practice, we can combine the essential results of all such pairings into one dataframe `shanghainese.bivariate` with again the function `nominal` and its `summary` method (N.B. excluding the outcome, TOPIC-MARKER, in column 1). Notice again that the notation of the association measures has been modified, from e.g. `uc.RC` and `uc.CR` to `uc.12` and `uc.21`, respectively, so that the number code '1' refers to the first (independent) variable, i.e. `category1` (corresponding to Row in the analysis above), and '2' to the second (independent) variable, i.e. `category2` (corresponding to Column in the analysis above):

```
> shanghainese.bivariate <- nominal(. ~ ., data=shanghainese.logical[-1])
435: [100][200][300][400]
> summary(shanghainese.bivariate)
Bivariate analysis of categorical variables:
Independents: TOPIC_LENGTH.1, TOPIC_LENGTH.2, TOPIC_LENGTH.3, TOPIC_LENGTH.4, TOPIC_LENGTH.5,
TOPIC_LENGTH.6, TOPIC_LENGTH.7, TOPIC_LENGTH.8, TOPIC_LENGTH.9,
TOPIC_LENGTH.10, TOPIC_POS.NOM, TOPIC_POS.ADJ, TOPIC_POS.ADV, TOPIC_POS.CLAUSE,
TOPIC_POS.VERB, FUNCTION.INTR, FUNCTION.COUNTER, FUNCTION.COND, FUNCTION.CONT,
FUNCTION.EMPH, COMMENT_TYPE.CLAU, COMMENT_TYPE.PHRA, COMMENT_TYPE.TAG,
COMMENT_TYPE.TSTR, COMMENT_TYPE.TSTRD, GENRE.MONO, GENRE.CONV, GENRE.INTER,
GENRE.SCRIPT


        category1       category2 N1  N2 N12       uc.12       uc.21
1  TOPIC_LENGTH.1  TOPIC_LENGTH.2 34 134   0  0.08899314  0.03803633
2  TOPIC_LENGTH.1  TOPIC_LENGTH.3 34  98   0  0.06210295  0.03118088
3  TOPIC_LENGTH.1  TOPIC_LENGTH.4 34  75   0  0.04621127  0.02715929
4  TOPIC_LENGTH.1  TOPIC_LENGTH.5 34  47   0  0.02803365  0.02234398
5  TOPIC_LENGTH.1  TOPIC_LENGTH.6 34  40   0  0.02367189  0.02109599
6  TOPIC_LENGTH.1  TOPIC_LENGTH.7 34  24   0  0.01395581  0.01800311
7  TOPIC_LENGTH.1  TOPIC_LENGTH.8 34  32   0  0.01877072   0.0196073
8  TOPIC_LENGTH.1  TOPIC_LENGTH.9 34  11   0 0.006307942  0.01482267
9  TOPIC_LENGTH.1 TOPIC_LENGTH.10 34   5   0 0.002849005  0.01263878
10 TOPIC_LENGTH.1   TOPIC_POS.NOM 34 195  20  0.02353412 0.008742737
... [ omitted 396 rows ] ...


Association measures:
  statistic.12 ~ statistic(1|2) ~ statistic(category1|category2) [ ~ statistic(R|C) ]
  statistic.21 ~ statustic(2|1) ~ statistic(category2|category1) [ ~ statistic(C|R) ]
```

Now, we can extract those pairings for which the asymmetric *Uncertainty Coefficient* $UC > .3$ in either direction, as follows (using the data frame `sumry.table` created with the `summary` method for the results of `nominal`), bringing forth only the following five cases:

```
> subset(summary(shanghainese.bivariate)$sumry.table, uc.12>.3 | uc.21>.3)
            category1           category2  N1  N2 N12      uc.12     uc.21
319     FUNCTION.INTR      FUNCTION.EMPH 197 167   0  0.3282683 0.3455473
371 COMMENT_TYPE.CLAU  COMMENT_TYPE.PHRA 356  56   0  0.2635471 0.4511873
372 COMMENT_TYPE.CLAU   COMMENT_TYPE.TAG 356  50   0  0.2317282 0.4279471
374 COMMENT_TYPE.CLAU COMMENT_TYPE.TSTRD 356  22   0 0.09550244 0.3177257
402         GENRE.MONO        GENRE.INTER 258 138   0   0.373133  0.438685
```

# 4 Multivariate analysis – polytomous logistic regression

## 4.1 Fitting a polytomous model

We start the multivariate analysis by fitting a polytomous logistic regression model with the function `polytomous` within the `polytomous` package, assigning the results to `shanghainese.polytomous`. First, however, we must remember to transform the variable TOPIC-LENGTH into numeric form:

```
> shanghainese$TOPIC_LENGTH <- as.numeric(shanghainese$TOPIC_LENGTH)
> shanghainese.polytomous <- polytomous(TOPIC_MARKER ~ TOPIC_LENGTH + TOPIC_POS +
+ FUNCTION + COMMENT_TYPE + GENRE, data=shanghainese)
```

A summary of the key results of the polytomous logistic regression model can be printed as follows, with the argument `max.print=NA` so that all the odds will be output (instead of only the first 10 lines which is the default setting):

```
> print(summary(shanghainese.polytomous), max.print=NA)
Formula:
TOPIC_MARKER ~ TOPIC_LENGTH + TOPIC_POS + FUNCTION + COMMENT_TYPE +
    GENRE
Heuristic:
one.vs.rest

Odds:
                        ne        a       mo       zi       ma
(Intercept)        0.08782   0.2628 0.009559    5.804   0.2585
COMMENT_TYPEPHRA   (0.568) (0.7328)   (1.42)  (1.283)   (1.42)
COMMENT_TYPETAG    (1.261) (0.5884)   (1.22)  (0.855) (0.9522)
COMMENT_TYPETSTR  (0.3588)  (2.456)  (1.632) (0.5519) (0.5734)
COMMENT_TYPETSTRD (0.8791)  (1.634) (0.6193) (0.4156)    (1.1)
FUNCTIONCOND         7.556   0.3025  (1.313)  0.07156  (1.055)
FUNCTIONCONT         12.87    0.281  0.09028  0.02395   2.697
FUNCTIONCOUNTER      9.425 (0.8001) (0.3065)   0.1788  (1.151)
FUNCTIONEMPH         2.833  (1.364) (0.6123)    0.525   (1.07)
GENRECONV           0.3315  0.03177    4.286 (0.8568)   4.046
GENREINTER         (0.8097)   0.1964    9.254  (1.043) (0.5909)
GENRESCRIPT        (0.2833)  (1.688)  (3.219) (0.4916)  (1.011)
TOPIC_LENGTH         1.185  (1.122)    1.437   0.5414   0.8104
TOPIC_POSADJ       (0.2623) (0.5769)    12.54   0.1827  (1.462)
TOPIC_POSADV        (1.629)  (1.119) (0.2926) (0.5171)  (1.307)
TOPIC_POSCLAUSE     0.2126 (0.9687)     2.99 (0.7534)  (2.192)
TOPIC_POSVERB       0.4591  (1.815)    2.709   0.2649  (1.875)

Null deviance:              1609  on  2500  degrees of freedom
Residual (model) deviance:  1191  on  2415  degrees of freedom
```

```
R2.likelihood:  0.26
AIC:            1361
BIC:            1719
```

Notice that one value per each categorical predictor variable is "missing". This is obligatory since the algorithm for fitting a logistic regression model cannot converge if all values of a categorical variable are included in the model (due to exact collinearity). However, the aggregate effects of these excluded, *default* or *reference* categories are represented jointly in the `Intercept` values. The `polytomous` and the underlying `glm` functions automatically select the first level/category of a factor as such a default value that will not receive odds/logodds estimates of their own. These default levels will normally be the alphabetically first ones for each factor, unless otherwise specified. Nevertheless, one can use the `relevel` or `reorder` functions to redefine these default levels. For the `shanghainese` data, we have thus selected as the default categories CLAUSE for COMMENT-TYPE, INTRODUCTORY for FUNCTION, MONOLOGUE for GENRE, and NOMINAL for PART-OF-SPEECH. As an example we could set EMPHATIC instead as the default category for FUNCTION as follows (this will not matter with respect to the results below since the polytomous model has already been fit):

```
> levels(shanghainese$FUNCTION)
[1] "INTR"    "COUNTER" "COND"    "CONT"    "EMPH"

> shanghainese$FUNCTION <- relevel(shanghainese$FUNCTION,"EMPH")
> levels(shanghainese$FUNCTION)
[1] "EMPH"    "INTR"    "COUNTER" "COND"    "CONT"
```

Returnung to the model, we can specifically extract various statistics concerning its fit, e.g. $R^2_L$ and *Accuracy*, as follows:

```
> shanghainese.polytomous$statistics$R2.likelihood
[1] 0.2599525

> shanghainese.polytomous$statistics$accuracy
[1] 0.5
```

The recall statistics for each topic marker can also be retrieved:

```
> shanghainese.polytomous$statistics$recall.predicted
  ne    a   mo   zi   ma
0.39 0.57 0.61 0.57 0.36
```

And we can get a crosstabulation of the predicted topic markers against the originally occurring ones:

```
> shanghainese.polytomous$statistics$crosstable
     ne  a mo zi ma
  ne 39 20 21 12  8
  a   8 57  8 23  4
  mo  6 16 61 11  6
  zi  6 23  6 57  8
  ma 17 22 11 14 36
```

## 4.2 Probability estimates and examplary sentences

The element fitted in the polytomous logistic regression model contains probability estimates for all topic markers in all the contexts/sentences in the original data that was used to fit the model:

```
> head(shanghainese.polytomous$fitted)
          ne          a          mo          zi        ma
1 0.38642999 0.01578725 0.409326035 0.001323448 0.1871333
2 0.10868093 0.26752516 0.170456081 0.181640481 0.2716974
3 0.04686899 0.22413264 0.333330553 0.182929151 0.2127387
4 0.53452365 0.18676178 0.001444502 0.111038765 0.1662313
5 0.52918897 0.04650779 0.025870388 0.197656422 0.2007764
6 0.16639150 0.30712478 0.020096350 0.299564515 0.2068229
```

We can scrutinize visually the overall distributions of the probability estimates, ranked within the sentences, by using the `plot` method for fitted `polytomous` model objects, as follows, producing *Figure 1* in Han et al. (in press):

```
> plot(shanghainese.polytomous, values="probabilities", panes="multiple")
```

The matrix of probability estimates allows us to do various operations, e.g. sorting the probabilities (horizontally) for each context/sentence (N.B. with the resultant matrix transposed, and the probability estimates thus in columns in increasing order, when `sort` is used without any arguments together with `apply`):

```
> apply(shanghainese.polytomous$fitted,1,sort)[,1:5]
               1         2          3           4          5
[1,] 0.001323448 0.1086809 0.04686899 0.001444502 0.02587039
[2,] 0.015787247 0.1704561 0.18292915 0.111038765 0.04650779
[3,] 0.187133282 0.1816405 0.21273867 0.166231308 0.19765642
[4,] 0.386429989 0.2675252 0.22413264 0.186761779 0.20077643
[5,] 0.409326035 0.2716974 0.33333055 0.534523647 0.52918897
```

Then, the resultant matrix contains e.g. the second highest and maximum probability estimates for each sentence on the fourth and fifth lines, respectively (with with the first five sentences, i.e. columns, selected below):

```
> apply(shanghainese.polytomous$fitted,1,sort)[4:5,1:5]
           1         2         3         4         5
[1,] 0.386430 0.2675252 0.2241326 0.1867618 0.2007764
[2,] 0.409326 0.2716974 0.3333306 0.5345236 0.5291890
```

This can be used to find out the number of cases/sentences for which the maximum and second highest probability estimates are both $P > 0.3$:

```
> length(which(apply(apply(shanghainese.polytomous$fitted,1,sort)[4:5,],2,function(x) all(x>0.3))))
[1] 154
```

The element fitted can also be manipulated vertically (by each topic marker) to provide us with the indices of those sentences which have received the highest probability estimates (for each topic marker); here we opt to see only the top five sentences for each topic marker:

```
> apply(shanghainese.polytomous$fitted, 2, function(x) order(x, decreasing=T))[1:5,]
```

```
      ne   a  mo  zi  ma
[1,] 164 185 243 470 222
[2,] 271 300 327 476 235
[3,]  50 176 354 478  40
[4,]   8 330 310 449 292
[5,] 282 122 356 472 215
```

Using the above indices, we can then extract the probability estimates for all five topic markers for the selected sentences #222, #185, and #40 discussed in the text.

```
> round(shanghainese.polytomous$fitted[222,],3)
   ne     a    mo    zi    ma
0.125 0.003 0.034 0.015 0.822
> round(shanghainese.polytomous$fitted[185,],3)
   ne     a    mo    zi    ma
0.072 0.688 0.095 0.035 0.109
> round(shanghainese.polytomous$fitted[40,],3)
   ne     a    mo    zi    ma
0.148 0.004 0.049 0.008 0.791
```

Finally, we can evaluate the dispersion of the topic markers for each sentence/context using the standard deviation (calculated with the standard function `sd`), which allows us to extract those sentences for which the dispersion is the smallest, suggesting that the probability estimates for all five topic markers are overall closest to each other:

```
> order(apply(shanghainese.polytomous$fitted,1,sd))[1:5]
[1]  94 296   2 102 365
```

Then, we can look up the probability estimates for the most equiprobable sentence #94:

```
> round(shanghainese.polytomous$fitted[94,],3)
   ne     a    mo    zi    ma
0.135 0.163 0.273 0.232 0.197
```