Regression analysis in translation studies

Stefan Th. Gries & Stefanie Wulff
University of California, Santa Barbara / University of North Texas, Denton

This paper provides an overview of how to compute simple binary logistic regressions and linear regressions with the open source programming language R on the basis of data from the INTERSECT corpus of English texts and their French and German translations. First, we show how one of the key statistics of logistic regressions is conceptually similar to the chi-square test of frequency tables. Second, we exemplify different applications of logistic regressions – with a binary predictor, with an interval/ratio-scaled predictor, and with a combination of both. Finally, we briefly exemplify a linear regression. In all cases, we discuss significance tests and provide examples for effective visualizations.

1. Introduction

1.1 Types of regressions and variables

One of the most remarkable current trends in theoretical and applied linguistics is the evolution of the field towards more empirically rigorous and quantitative methods. In theoretical linguistics, after a long reign of generative approaches to grammar and their largely intuitive grammaticality judgments, there is now a lot of interest in, and work on, probabilistic theories of language acquisition, representation, and processing, and such approaches rely more and more on experimental and observational data that are analyzed with statistical tools. In a similar vein, many areas of applied linguistics such as second language acquisition also have turned to quantitative tools, and translation studies are no exception.

Given that this is only a relatively recent development and that different kinds of data are only slowly becoming available (e.g. corpora on lesser-studied languages and/or parallel and aligned corpora), the move towards more quantitative methods is still in progress. Practitioners are constantly learning about, and developing, new methods and areas of application for existing methods. One particularly flexible and widespread family of methods is that of *regression analysis*. This method involves analyzing the degree to which a *dependent*

variable is correlated with one or more *predictors*, where we use predictors as a cover term for both individual *independent variables* and their n-*way interactions*. The dependent and independent variables in a regression can be of various levels of measurement:

- *categorical data*, i.e. data that reflect that data points belong to different categories such as a binary variable ClauseOrder ('main clause' vs. 'subordinate clause') or an *n*-ary variable such as PhraseType (NP vs. VP vs. PP);
- *ordinal data*, i.e. rank-ordered data such as syntactic Complexity (on a scale such as "high" > "intermediate" > "low");
- *ratio-/interval-scaled*, i.e. continuous numeric data such as SyllableLength, ReactionTime, etc.

At the risk of some simplification, regressions are distinguished depending on (i) the type of relation between the dependent variable and the predictors and (ii) the level of measurement of the dependent variable. As to (i), one can distinguish between *linear regressions* and *non-linear regressions*; we will only focus on the former. As to (ii), one can distinguish between *binary logistic* and *multinomial logistic regressions* (for categorical dependent variables), *ordinal/multinomial regressions* (for ordinal data), and *linear regressions* (for ratio-/interval data).

Often but not necessarily, the dependent variable can be conceptualized as the effect, whereas the predictors can be conceptualized as causes. Regression analyses are typically used to compute expected values of a dependent variable, which allow to predict numeric, or classify categorical, values of dependent variables. In this chapter, we will discuss binary logistic regression and linear regression. For mathematical reasons, linear regressions would usually be introduced first, but given that (i) binary logistic regression can be shown to be related to the χ^2-test (chi-squared test) many scholars are familiar with and (ii) data in translation studies are probably less likely to be of a type that allows linear regressions, we will not follow this usual pattern. Note also that a zip-file with example data and code is available from the first author's website at ⟨http://tinyurl.com/stgries⟩.

1.2 The example data

The data to be used to exemplify regressions here are from the INTERSECT corpus compiled and graciously provided by Raf Salkie at the University of Brighton. The corpus consists of

Table 1. Schematic excerpt of the raw data table for German in case-by-variable format

Case	Preceding	Match	Subsequent	Order	SubordType	Len_ MC	...
1	...	weil	...	mc-sc	causal	9	...
2	...	nachdem	...	sc-mc	temporal	7	...
...	...	...	...	...	...	...	...

1.3 The software

These days, statistical analyses are done computationally. There are many applications available and for many reasons, we are using R (R Development Core Team 2011). R is not just a statistics program, but a full-fledged programming language, which entails that it does not by default come with a nice point-and-click GUI, but a command-line interface. However, it is freely available software, the leading platform for the development and implementation of new statistical techniques, and, given its open-source nature, immensely powerful in terms of the number and range of methods and graphs available.

When R is started, by default it only shows a fairly empty console and expects user input from the keyboard. Nearly all of the time, the input to R consists of what are called functions and arguments. Just like in a spreadsheet software, *functions* are commands that tell R what to do; *arguments* are specifics for the commands, namely what to apply a function to (e.g. a value/number, the first row of a table, a complete table, etc.) or how to apply the function to it (e.g. whether to compute a mean or a logarithm, which kind of logarithm to compute: a binary log, a natural log, etc.).

Before we explore how to understand and perform regressions, we first need to load the data into R. One way to read a raw data file involves the function read.delim, which, if the raw data table has been created as outlined above, requires only one argument, namely file, which, when defined as below, prompts the user to choose the path to the file containing the data; crucially, the two files are aligned such that the *n*th row in the English file is the translation of the *n*th row in the German file. The following code will therefore load the files with the German and the English data into R, where the "<–" tells R to store content into the data structure to the left of the 'arrow' (i.e. here a *data frame*, R's version of a table, german and english), where "¶" means 'press ENTER', and where text after a # is ignored and can be used for comments:

```
german <- read.delim(file = file.choose()) # load German data into german¶
english <- read.delim(file = file.choose()) # load English data into english¶
```

To check whether the loading was successful, we can explore the data frame. The function summary summarizes each column of the data frame by either

listing the most frequent levels (for categorical variables) or by providing numerical summaries such as minima, maxima, means, etc. (for numerical variables):

```
summary(german)¶
      CONJ       SUBORDTYPE    ORDER         LEN_MC          LEN_SC
als    : 93   caus: 199    mc-sc: 275   Min.   :  2.000   Min.   :  2.000
bevor  : 46   temp: 204    sc-mc: 128   1st Qu.:  6.000   1st Qu.:  5.000
nachdem: 65                             Median :  8.000   Median :  8.000
weil   :199                             Mean   :  9.266   Mean   :  9.362
                                        3rd Qu.: 11.000   3rd Qu.: 12.000
                                        Max.   : 31.000   Max.   : 36.000
```

The next section will explain aspects of the logic underlying logistic regressions. For reasons of space, this chapter can of course not provide a comprehensive introduction to all its details and complications; Section 4 will mention some useful references for follow-up study.

2. Methods 1: Binary logistic regression

Logistic regression is a regression method that does not come easy to beginners. This is because, unlike the default type of linear regression discussed later, it involves a transformation of the data that ensures that the regression predicts values that are theoretically plausible and/or practically possible to attain. However, at least a first understanding can be gained by comparing the results of a logistic regression with the more familiar χ^2 test and the related G statistic.

2.1 From cross-tabulation to binary logistic regression

To determine, for instance, whether there is a tendency for causal and temporal subordinate clauses to prefer a particular clause order in German (and later in English), a first descriptive step would be a cross-tabulation. In R, this can be done easily with the function table, which only requires the two variables (vectors or factors, in R's parlance) as arguments; the first named argument goes into the rows, the other into the columns. The result of the tabulation is assigned to a data structure orders and then printed to the screen:

```
orders <– table(german$SUBORDTYPE, german$ORDER)¶
orders¶
       mc-sc  sc-mc
caus     184     15
temp      91    113
```

There seems to be a strong correlation such that German causal subordinate clauses prefer to occur after main clauses whereas German temporal subordinate clauses prefer to occur before main clauses. It is usually useful to (i) compare the observed frequencies in against those expected by chance from the row and column totals and (ii) quantify this comparison by means of the so-called Pearson residuals. These residuals are positive (or negative) if the observed frequencies are larger (or smaller) than the expected frequencies, and the more they deviate from zero, the stronger the effect. In R, we can compute these easily from the results of a χ^2-test as computed with the function chisq.test, which requires as arguments the table to be tested (i.e. orders) and correct = FALSE (when the sample size $n > 60$):

```
test.orders <– chisq.test (orders, correct = FALSE)¶
```

Now, we can retrieve the residuals of the four cells, which are computed as shown in (3) and which reveal the strong pattern already suggested by the observed frequencies above:

(3) $$\frac{observed - expected}{\sqrt{expected}}$$

```
test.orders$expected # compare top left to (199*275)/403¶
             mc-sc      sc-mc
caus      135.7940   63.20596
temp      139.2060   64.79404
test.orders$residuals # compare top left to (184–135.794)/sqrt(135.794)¶
             mc-sc      sc-mc
caus       4.136760  –6.063476
temp      –4.085750   5.988708
```

The χ^2-value from a χ^2-test is the sum of the squared residuals, as shown in (4), and here, the preferences of the subordinate clause types are highly significant, as shown by the p-value:

(4) $$\chi^2 = \sum \frac{(observed - expected)^2}{expected}$$

```
test.orders # compare to sum (test.orders$residuals^2)¶
        Pearson's Chi-squared test
data: orders
X-squared = 106.4365, df = 1, p-value < 2.2e-16
```

One important way to quantify the size of this highly significant correlation is the *odds ratio*, which tells you how the likelihood of one variable level changes in response to how the other variable changes. Here, for causal clauses, the *odds* for "mc-sc" are 184/15≈12.267 (i.e. this is how much "mc-sc" is more likely than "sc-mc"), for temporal clauses the odds are 91/113≈0.805 (i.e. this is how much "mc-sc" is more likely than "sc-mc" – i.e. it is *less* likely). Thus, looking at both clause types, the odds ratio for "mc-sc" is 12.267/0.805≈15.23 times more likely with causal clauses than with temporal clauses. Often, this odds ratio is logged to yield *log odds* so that the range of possible values extends nicely from $-\infty$ to $+\infty$ (with 0 reflecting the absence of a correlation).

```
(184/15) / (91/113) # odds ratio¶
[1] 15.23223
log (15.23223) # log odds¶
[1] 2.723414
```

While the χ^2-test/value is widely used, another way to test such distributions for significance is the *G*-test/value. It is computed as shown in (5) and below:

$$(5) \qquad G^2 = 2 \cdot \sum observed \cdot \ln\left(observed / expected\right) \left(\text{where ln} = \text{'natural logarithm'}\right)$$

```
2*sum(orders*log(orders/test.orders$expected)) # G¶
[1] 116.9747
```

Most of the time, the results of a χ^2-test and the *G*-test are very similar, but it is the latter that is very frequently used in the context of this chapter's topic, regression modeling. More specifically, the notion underlying regressions is that outcomes of a *dependent variable* (or *response*, often considered an 'effect') are modeled as a function of one or more *predictors* (i.e. *independent variables* and/or their interactions, often considered 'causes') in a regression equation. To determine which predictors are needed to predict the dependent variable in a way that strikes a balance between prediction accuracy and Occam's razor, such a regression modeling process involves comparing different models to each other. One of these models is the so-called *null model*, i.e. a model without any predictors (reflecting just the two orders' overall frequencies). In the simplest case, one compares a model with one predictor (such as german$SUBORDTYPE) against the null model, and if the one predictor makes significantly better predictions than the null model, we say it has a significant effect; it should become clear that this approach is largely only terminologically different from the simple χ^2-test. Let's apply this approach to the present question to better appreciate the analogy.

2.2 Binary logistic regression with one binary predictor

As a first step, we load the package Design into R (because it makes some aspects of regressions easier than R's standard functionality). Then, we define a logistic regression model (lrm) model.01 using a formula, which contains the response, a tilde, and the predictor(s) as well as the argument data to tell R where the variables come from. Then we print this model (only parts of the output will be provided and discussed here).

```
library(Design)¶
model.01 <- lrm(ORDER ~ SUBORDTYPE, data = german)¶
model.01¶
[...]
         Obs    Max       Deriv     Model L.R.         d.f.          P
         403              2e-09         116.97            1          0
           C                Dxy        Gamma         Tau-a
       0.776              0.552        0.877          0.24
          R2              Brier
       0.353              0.159
                                 Coef      S.E.      Wald  Z  P
Intercept                      −2.507    0.2685    −9.34     0
SUBORDTYPE = temp               2.723    0.3032     8.98     0
```

The output is best approached with three questions. First, "is there a significant correlation between the response and the predictor(s)?" Yes, there is: among other things, the output contains (in the rectangle) the above *G*-value (as Model Likelihood Ratio), the above degrees of freedom (d.f.), and the *p*-value (0). Thus, the model with one predictor (german$SUBORDTYPE) fares significantly better than the null model and we say the predictor is significantly correlated with the clause order.

Second, "how well does the model predict clause orders?" [1] This is answered by the circled *C*- and R^2-values provided by R. *C* ranges from 0.5 (predictions are at chance accuracy) to 1 (perfect predictive accuracy; ideally, $C \geq 0.8$), and R^2 is a particular version of a coefficient of determination ranging from 0 (no correlation between the response and the predictors) and 1 (perfect correlation between the predictors). Here, *C* is not quite high enough, but we are very close to it so the model's accuracy is 'not bad'.

1. For reasons of space, in this paper, we do not concern ourselves with the difference between predicting data and classifying data.

Third, "if there is a significant effect, what is its nature?" For this question, we have to turn to the table at the bottom and the coefficients in the rounded rectangle, but also understand what a regression equation does. In the case of logistic regression, R tries to define an equation that computes the probability of the alphabetically second level of the response (i.e. "sc-mc"), and it does that using

– an *intercept*, which reflects the probability of "sc-mc" when all other categorical predictors are their alphabetically first reference levels (i.e. when SubordType = "caus") and/or all interval-/ratio-scaled predictors (none here) are zero;
– a *coefficient* for each predictor's effect on the probability of "sc-mc"; here, a coefficient for when SubordType changes from "caus" to "temp".

That is to say, when the (sole) predictor is "caus", the regression equation becomes (6) (because, since SubordType = "caus", the coefficient for SubordType = "temp" 'does not apply' and is set to 0). On the other hand, when the (sole) predictor is "temp", then the regression equation's result does apply and is set to 1, yielding (7).

(6) regression result = −2.507 + 0·2.723 = −2.507

(7) regression result = −2.507 + 1·2.723 = 0.216

But what do these results mean and how can they reflect probabilities when they are not between 0 and 1 (as probabilities are)? The answer to both questions is that these regression results are so-called *logits* of the probabilities that the regression predicts (as shown in (8)), which means we can get the predicted probabilities by computing the *inverse logits* (as shown in (9)).

(8) logit of a probaility $p = \log \frac{p}{1-p}$

(9) inverse logit of a value x $= \frac{1}{1+e^{-x}}$

Thus, if we apply (9) to (6) and (7), our third question gets answered: we see how much SubordType = "temp" increases the probability of "sc-mc" (compared to SubordType = "caus"):

– the predicted probability of "sc-mc" when SubordType = "caus" is ≈ 0.075;

```
1/(1 + exp (−−2.507))¶
[1] 0.07536891
```

– the predicted probability of "sc-mc" when SUBORDTYPE = "temp" is ≈ 0.554.[2]

```
1/(1 + exp (–0.216))¶
[1] 0.553791
```

To sum up, in the German data, there is a highly significant and intermediately strong correlation between the type of subordinate clause and the position it prefers to occupy relative to the main clause ($R^2 = 0.353$, $G = 116.97$, $df = 1$, $p < 0.001$): causal subordinate clauses prefer to follow the main clause whereas temporal subordinate clauses prefer to precede it. Ideally, the reader would perform this type of analysis for the English data provided in the companion file and find that (i) in the English data, the order "sc-mc" is much more frequent than in the German translations and (ii) while temporal subordinate clauses in English also prefer "sc-mc" more than causal clauses, that preference is less strong.

2.3 Binary logistic regression with an interval-/ratio-scaled predictor

In this section, we will turn to an interval-/ratio-scaled predictor, namely LENGTHDIFF. Note that this variable is the difference of main clause length minus subordinate clause length (in words). Thus, when that value is positive, the main clause is longer than the subordinate clause, and when it's negative, the main clause is shorter. We fit a new model.01, overwriting the one from the previous section:

```
model.01 <- lrm(ORDER ~ LENGTH_DIFF, data = german)¶
model.01¶
[...]
          Obs    Max   Deriv    Model      L.R.      d.f.         P
          403          2e-13                7.58         1    0.0059
            C            Dxy            Gamma      Tau-a
        0.603          0.207
           R2          Brier
        0.217           0.09    0.026    0.213
                    Coef          S.E.      Wald    Z  P
Intercept          –0.77673      0.10849   –7.16      0.000
LENGTH_DIFF         0.04418      0.01639    2.69      0.007
```

2. Another way to interpret the coefficients is to antilog them: First, $e^{\text{intercept}} = e^{-2.507} = 0.08151$, which one recognizes are the 15/184 odds for "sc-mc" when SUBORDTYPE = "temp". Second, $e^{\text{coefficient for SUBORDTYPE = "temp"}} = e^{2.723} = 15.22593$, which one recognizes is the above odds ratio (cf. p. 41).

We can analyze this output with the same three questions as above. First, there is a significant correlation between the clause order in German and the length difference between clauses; however, the correlation is considerably weaker than before ($R^2 = 0.217$, $G = 7.58$, $df = 1$, $p = 0.0059$). Second, in line with the weaker correlation, the model's accuracy at predicting the right order is also markedly worse (R^2: see above, $C = 0.603$). Third, the nature of the effect: As mentioned above, the intercept reflects the probability of "sc-mc" when all other categorical predictors are their alphabetically first reference levels [none here] and/or all interval-/ratio-scaled predictors are zero. Thus, the probability of "sc-mc" when LengthDiff = 0 (both clauses are equally long) is this:

```
1/(1 + exp (–0.77673))¶
[1] 0.3150251
```

The coefficient of LengthDiff, on the other hand, reflects how the probability for "sc-mc" changes for every unit-change of LengthDiff, i.e. when, for example, LengthDiff is not 0 but 1, is not 1 but 2, etc. Most importantly, the fact that it is positive shows that, as LengthDiff increases – i.e. as main clauses get longer compared to their subordinate clauses – the probability that they occur after the subordinate clause ("sc-mc") increases, too. And given an overall tendency of short-before-long in many ordering phenomena, that makes a lot of sense. It is crucial, however, to realize that, because the coefficient for LengthDiff is used to compute logits of probabilities and the logit transformation is non-linear, changes in LengthDiff do not affect the probability of "sc-mc" uniformly. For instance, the code below shows that the probability of "sc-mc" does not always increase by the same value when the subordinate clause becomes a word longer.

```
1/(1 + exp (–(–0.77673 + (–20*0.04418))))¶
[1] 0.1597177
1/(1 + exp (–(–0.77673 + (–19*0.04418))))¶
[1] 0.1657365 # a 1-word increase (from –20 to –19) of the length difference
      increases p("sc-mc") by 0.0060188
1/(1 + exp (–(–0.77673 + (–10*0.04418))))¶
[1] 0.2281952
1/(1 + exp (–(–0.77673 + (–9*0.04418))))¶
[1] 0.2360696 # a 1-word increase (from –10 to 9) of the length difference
      increases p("sc-mc") by 0.0078744
1/(1 + exp (–(–0.77673 + (0*0.04418))))¶
[1] 0.3150251
1/(1 + exp (–(–0.77673 + (1*0.04418))))¶
[1] 0.3246354 # a 1-word increase (from 0 to 1) of the length difference
      increases p("sc-mc") by 0.0096103
```

This effect is also illustrated in Figure 1 and the main reason that logistic regressions are sometimes difficult to understand. Note in particular how the right panel shows how the logit transformation makes the predicted probabilities level off close to $y = 0$ and $y = 1$ so that no probabilities < 0 or > 1 can be predicted. Given the difficulty of understanding such results, one will sometimes find a so-called *average predicted difference*, which gives "the expected, or average, difference in [the predicted probability] corresponding to a unit difference in [an input variable]" (cf. Gelman & Hill 2008: 101ff.), but other alternatives are also available. Again, ideally, the reader would perform the analogous analysis for the English data and find that (i) the order "sc-mc" is of course still much more frequent in the English than in the German data, but (ii) in the English translations, LengthDiff has no significant effect on the clause ordering.

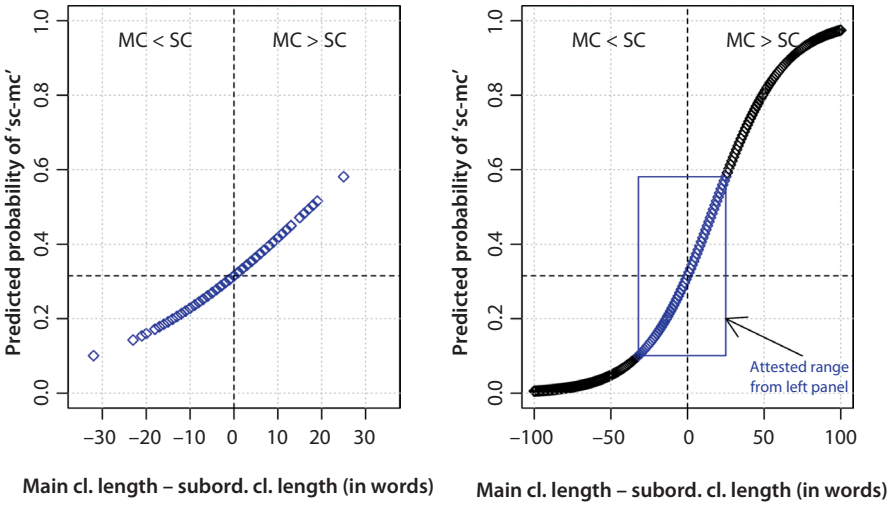

Figure 1. The probability of "sc-mc" as a function of LengthDiff

2.4 Logistic regression with more than one predictor

Let us finally look at one more complex example, a logistic regression that involves a categorical predictor, an interval-/ratio-scaled predictor, and their interaction. In this example, we try to predict the ordering in German on the basis of the subordinating conjunction that is used – three temporal ones (*als* 'as'/'when', *bevor* 'before', and *nachdem* 'after') and the causal *weil* 'because' – and LengthDiff from above as well as their interaction. We again define a model object and the only new point is that predictors to be included together with their interaction are combined with "*" (if the interaction is not wanted, one uses a "+" instead):

```
model.01 <- lrm (ORDER ~ CONJ*LENGTH_DIFF, data = german)¶
model.01¶
[...]
```

Obs	Max	Deriv	Model	L.R.	d.f.	P
403		6e-09		135.21	7	0
C		Dxy		Gamma	Tau-a	
0.818		0.636		0.674	0.276	
R2		Brier				
0.399		0.149				

	Coef	S.E.	Wald	Z	P
Intercept	0.46337	0.22543	2.06		0.0398
CONJ = bevor	−0.81963	0.39463	−2.08		0.0378
CONJ = nachdem	−0.06495	0.34115	−0.19		0.8490
CONJ = weil	−2.96835	0.35207	−8.43		0.0000
LENGTH_DIFF	0.11171	0.03827	2.92		0.0035
CONJ = bevor * LENGTH_DIFF	−0.14790	0.06384	−2.32		0.0205
CONJ = nachdem * LENGTH_DIFF	−0.06980	0.05222	−1.34		0.1813
CONJ = weil * LENGTH_DIFF	−0.10948	0.05521	−1.98		0.0474

As before, it is best to approach the output with the three above questions in mind. First, there is a significant correlation between the response and the predictors, and it is the highest and strongest we have seen so far ($R^2 = 0.399$, $G = 135.21$, $df = 7$, $p < 0.001$). Second and correspondingly, the model does a good job at predicting the clause ordering (R^2: see above, and $C = 0.818$).

The third question – how to interpret the coefficients – is a bit harder to tackle in multifactorial models. This is for two reasons. First, in a multifactorial model, there can be an overall significant correlation (as indicated by R^2 etc.), but some predictors in the model may not contribute significantly to it. There are two conceptually very different ways of handling this. One is to perform what is called model selection: on the basis of Occam's razor, predictors that do not contribute enough to the model are weeded out successively until a model is found that contains only significant predictors (cf. Crawley 2007: Chapters 9, 17 for discussion). The other is to not perform model selection and report the insignificant predictors as insignificant (cf. Harrell 2001: Section 4.3 for discussion). Given space constraints, we must restrict ourselves to mentioning that the interaction of Conj and LengthDiff is only marginally significant ($p = 0.084$; cf. the companion code file) and only explain the for now most important issue, namely how to make sense of the coefficients.

The first part of interpreting coefficients is as discussed above. As before, the intercept reflects the predicted probability of "sc-mc" "when all other categorical predictors are their alphabetically first reference levels […] and/or

all interval-/ratio-scaled predictors are zero." Thus, the intercept here indicates the predicted probability when Conj = "als" and LengthDiff = 0:

```
1/(1 + exp (–0.46337))¶
[1] 0.6138133
```

Also as before, the coefficients for the other three conjunctions indicate the predicted probabilities of "sc-mc" when Conj ≠ "als" but another conjunction. It is immediately obvious that the only causal conjunction comes with a much lower probability of "sc-mc" ordering.

```
1/(1 + exp (–(0.46337 – 0.81963))) # bevor = sign. different from als¶
[1] 0.4118652
1/(1 + exp (–(0.46337 – 0.06495))) # nachdem ≠ sign. diff. from als¶
[1] 0.598308
1/(1 + exp(–(0.46337 – 2.96835))) # weil = sign. diff. from als¶
[1] 0.0755098
```

The coefficient for LengthDiff now indicates how the probability for "sc-mc" changes for every unit-change (word-length difference) of LengthDiff, when, crucially, Conj = "als". The following are the predicted probabilities of "sc-mc" when LengthDiff = 1 and 10 and when Conj = "als". As before, the coefficient is positive: as LengthDiff increases so does the probability that the subordinate clause with *als* precedes the main clause.

```
1/(1 + exp(–(0.46337 + 0.11171)))¶
[1] 0.6399345
1/(1 + exp(–(0.46337 + 10*0.11171)))¶
[1] 0.829271
```

The more interesting part is now concerned with the interaction of Conj and LengthDiff. For example, we have seen that increasing values of LengthDiff increase the probability of "sc-mc" for Conj = "als", but the (marginally significant) interaction now reveals that this is not so for the other conjunctions: When Conj = "bevor", then increasing values of LengthDiff decrease the probability of "sc-mc": we add 0.11171 for every word-length difference but must also subtract 0.1479 for every word-length difference. When Conj = "nachdem", then increasing values of LengthDiff decrease the probability of "sc-mc" compared to when Conj = "als": one adds 0.11171, but also has to subtract 0.0698, for every word-length difference. Finally, LengthDiff has hardly no effect when Conj = "weil": the 0.11171 that are added for each word more are nearly offset

completely by the –0.10948 that must be subtracted again. In other words, LengthDiff has different effects for each conjunction, which is graphically represented in Figure 2.

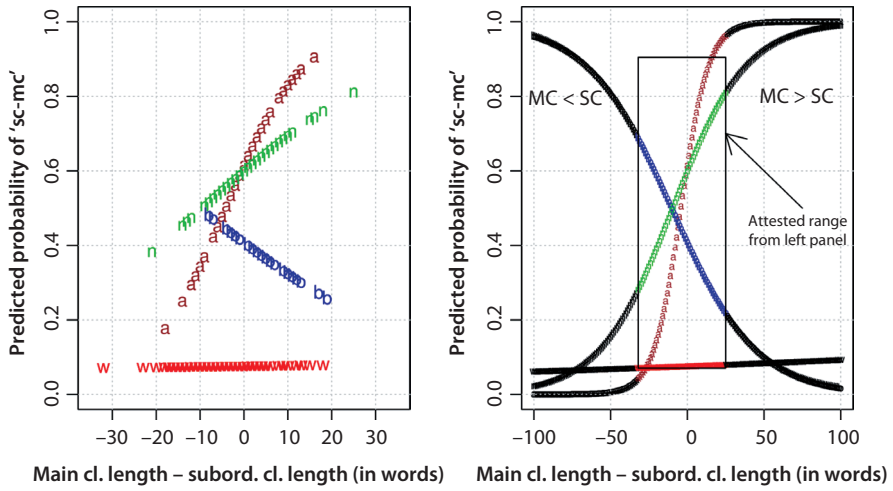

Figure 2. The probability of "sc-mc" as a function of Conj:LengthDiff

3. Methods 2: Linear regression

Let us now briefly turn to linear regressions. Just as logistic regressions are related to χ^2- and *G*-tests, so are linear regressions related to the product-moment correlation *r* and the *t*-test. If we want to determine how well we can predict the differences in clause lengths in English on the basis of the differences in clause lengths in the German translations, we use the function lm (for linear model) and then print its summary; note the use of the $-sign in dataframe$column

```
model.01 <- lm(english$LENGTH_DIFF ~ german$LENGTH_DIFF)¶
summary(model.01)¶
[...]
Coefficients:
                        Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)             −0.59386   0.32295      −1.839    0.0667.
german$LENGTH_DIFF       0.73522   0.04719       15.581   <2e-16 ***
[...]
Multiple R-squared:          0.3771,   Adjusted R-squared: 0.3755
F-statistic: 242.8 on 1 and 401 DF, p-value: < 2.2e-16
```

There is a significant correlation between the response and the predictor (adj. $R^2 = 0.3755$, $F_{1, 401} = 242.8$, $p < 0.001$), but the correlation is only intermediately high. Fortunately, the coefficients are much easier to interpret because a linear regression does not involve a (logit) transformation – it models the response directly. That means, the coefficient for the German LengthDiff indicates directly how much the predicted English LengthDiff increases for every unit increase of a German LengthDiff: 0.7355. The following two lines of code exemplify two such predictions:

```
−0.59386 + −32*0.73522 # prediction for when german$LENGTHDIFF = −32¶
[1] −24.1209
−0.59386 + 25*0.73522 # prediction for when german$LENGTHDIFF = 25¶
[1] 17.78664
```

The companion code file provides the code to produce Figure 3, which illustrates the correlation between the length differences and indicates the predicted trend with a regression line. In fact, the coefficient for German LengthDiff is the slope of the regression line. It is easy to see that the first prediction (cf. the left arrow) is quite good (because when $x = -32$, then the corresponding y-value is fairly close to −24) and that the second prediction (cf. the right arrow) is quite bad (because when $x = 25$, then the corresponding y-value is fairly far away from 17.8).

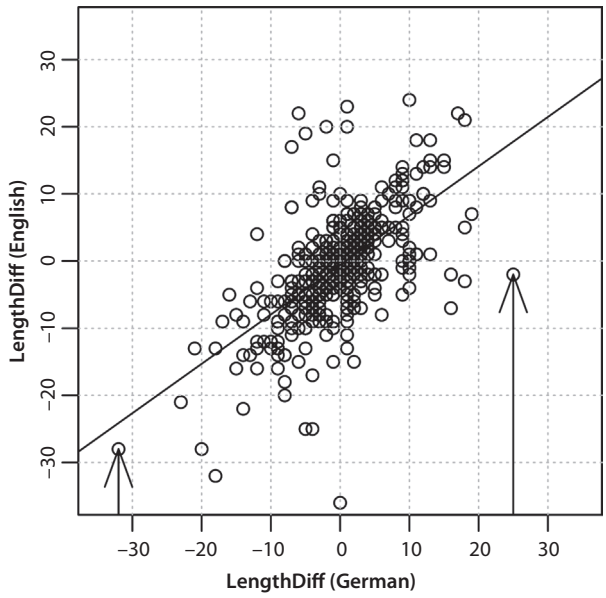

Figure 3. The regression from the German length differences to the English ones

Just like logistic regressions, linear regressions can involve more than one predictor, and the overall logic is the same as before: interactions between predictors, for example, would be reflected in different slopes of regression lines, which may or may not be significantly different from each other.

4. Concluding remarks

The application potential of logistic regression in translation studies is just as wide as it is for language research at large – in particular, the parallels between *translationese* and *interlanguage* in second language acquisition are more than obvious. That is, logistic regression is a most suitable tool whenever we assume

- that the translator's choice for a particular word or structure was determined by more than one independent variable; and/or
- that the translator was presented with more than one realization of wordings or structure in the source language; and/or
- the translator chose between more than one alternative word or structure in the language being translated into.

In the little case study presented here for illustrative purposes, all three situations applied: the ordering of main and subordinate clauses is generally seen as being multifactorially determined, and both orderings are possible in both the source language (German) and the translated language (English). Other possible applications of logistic regression (leaving it at German and English as example languages here) could be the analysis of such alternations at various levels of linguistic granularity, such as prenominal adjective ordering from, say, German (*der rote grosse Ball* vs. *der grosse rote Ball*) to English (the *red big ball* vs. *the big red ball*) or the other way around; the realization of the genitive from English (*Stefan's book* vs. *the book of Stefan*) to German (*Stefans Buch* vs. *das Buch vom Stefan*) or the other way around; or the variable realization of the German infinitival complement structure (*Steffi fing an zu kochen*) in English, where the translator has to make a choice between infinitival complements (*Steffi began to cook*) or gerundial complements (*Steffi began cooking*). Likewise, logistic regression could be employed to topics as diverse as synonym choice (English: *He was happy* ↔ German: *Er war froh/glücklich/munter/freudig erregt/...* or German: *Er war froh* ↔ English: *He was happy/glad/chipper/in a good mood/...*), optional complementizer realization (English: *I think that the movie is great/I think the movie is great* ↔ German: *Ich glaube, dass der Film gut ist/Ich glaube, der Film ist gut*), or attended/unattended demonstratives (English: *This paper has shown.../This has shown...* ↔ German: *Dieser Artikel hat gezeigt.../Dies hat gezeigt...*), to give but a few examples. All these phenomena

have been argued to be multifactorially determined in native language, and more recent work confirms that to be true also in second language acquisition; however, to our knowledge at least, no such studies exist yet on translated language.

While reasons of space preclude a more detailed discussion, it should be obvious by now that such regression approaches are a very powerful tool that can help uncover patterns that are interesting and would escape the naked eye (or trained intuition). With power come challenges, so we strongly encourage the interested reader to explore this methodology further. Recently, several statistics textbooks for linguists (using R) have been published, all of which cover different regression approaches: Baayen (2008), Johnson (2008), and Gries (2009) are good starting points, as is the *very* useful Pampel (2000). More general introductions to statistics and regression are Harrell (2001), Crawley (2007), Gelman & Hill (2008), and Hilbe (2009), and while there certainly is a learning curve, the power of such tools and their implications for linguistic research should make it worth to anybody with a serious interest in rigorous empirical research.

References

Baayen, R. Harald. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics Using R*. Cambridge: CUP.

Crawley, Michael J. 2007. *The R book*. Chichester: John Wiley and Sons.

Diessel, Holger. 2005. Competing motivations for the ordering of main and adverbial clauses. *Linguistics* 43: 449–470.

Diessel, Holger. 2008. Iconicity of sequence: A corpus-based analysis of the positioning of temporal adverbial clauses in English. *Cognitive Linguistics* 19: 457–482.

Gelman, Andrew & Hill, Jennifer. 2008. *Data Analysis Using Regression and Multilevel/Hierarchical models*, 2nd printing with corrections. Cambridge: CUP.

Gries, Stefan Th. 2009. *Statistics for Linguistics Using R: A Practical Introduction*. Berlin: Mouton de Gruyter.

Harrell, Frank E. Jr. 2001. *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. New York NY: Springer.

Hilbe, Joseph M. 2009. *Logistic Regression Models*. Boca Raton FL: Chapman & Hall.

Johnson, Keith. 2008. *Quantitative Methods in Linguistics*. Oxford: Blackwell.

Pampel, Fred C. 2000. *Logistic Regression: A Primer*. Thousand Oaks CA: Sage.

R Development Core Team. 2011. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ⟨http://www.R-project.org/⟩

Salkie, Raf. 1995. INTERSECT: A parallel corpus project at Brighton University. *Computers and Texts* 9: 4–5.