METHOD NOTES
CORPUS-BASED TRIGGERS ANALYSIS

Diana Carter

3 May 2011

1 Definitions

Trigger words: words that overlap both in form and meaning in the two languages. They include bilingual homophones and proper nouns, "allowing for small differences in phonological form".

The language of a trigger word cannot be determined, and thus is never counted as a codeswitch. Trigger words can belong to functional and lexical categories. Mirjam later analyzed the trigger items as: proper nouns, content words, and function words.

Basic clause: contains maximally but not minimally one main verb.

Codeswitch at the clause level: "a basic clause is considered to contain a codeswitch when it contains words from two languages or when it contains words from a language different than that in the previous basic clause. No language is assigned to the basic clause as a whole, but it is assessed whether a clause contains non-trigger words from either language". The codeswitch can therefore be intra-clausal or inter-clausal.

Codeswitch at the word level: assessed in a linear way. "A word is considered to be codeswitched when it differs in language from the previous word, regardless of their grammatical relation, and regardless of when a switch back occurs". "A word was considered to be codeswitched when it was part of a different language than the previous non-trigger word." (p455, emphasis added)

2 Clause-level analysis

The data should be divided into basic clauses. For every basic clause, determine if it contains one or more trigger words, and if it contains a codeswitch. There is a codeswitch if there are words from both languages in the same clause (intra-clausal), or if it contains words from a language different from the previous basic clause (inter-clausal).

The algorithm, set out in Figure 1, will output a series of 4 figures for statistical analysis.

Run the above with progressively looser constraints on T:

1. cognate nouns
2. cognate pluralised

```
split into clauses
mark any Ts (trigger-words) in each clause
provided the clause does not contain +<:
    look at the previous clause and get the language of the last non-T in that
        clause (call it P)
for clauses containing T:
    find the first non-T word (call it F)
        if language of F != language of P, count the clause as interclausal_plusT
    scan between F and clause-end
        if the segment contains one or more words whose language!= language of F,
            count the clause as intraclausal_plusT
for clauses not containing T:
    find the first word (F)
        if language of F != language of P, count the clause as interclausal_minusT
    scan between F and clause-end
        if the segment contains a word whose language != language of F, count the
            clause as intraclausal_minusT
```

Figure 1: Analysis algorithm for clauses

3. proper nouns
4. -[i]o verbs
5. adjectives
6. _[i]o verbs
7. IMs

Mirjam: At the clause level it would for example be the number of basic clauses:

1. with cognate + switched + intra-clausal
2. with cognate + not switched + intra-clausal
3. without cognate + switched + intra-clausal
4. without cognate + not switched + intra-clausal
5. with cognate + switched + inter-clausal
6. with cognate + not switched + inter-clausal
7. without cognate + switched + inter-clausal
8. without cognate + not switched + inter-clausal

3 Word-level analysis

The data should be divided into turns. Each turn is defined as a stretch of speech where the speaker did not interrupt his/her flow even if there was input or vocalisation from the other speaker. Where the speaker halts as a result of interjections from the other speaker, that is the end of that turn.

The turn starts at the first non-trigger word. Determine whether each non-trigger word follows or precedes a trigger word, and whether it is codeswitched. The word is codeswitched if it is part of a different language than the previous non-trigger word.

The algorithm, set out in Figure 2, will output a series of 2 figures for statistical analysis.

When T-1, T-2 or T+1 are themselves trigger words, count T as a switch if it is in a different language compared to the I nearest non-triggerword.

```
split each utterance into segments at each T
for each T, get the language of T-1, T-2, and T+1
compare T-1 and T-2
    if they are different languages, count the instance as pre-T codeswitch
compare T-1 and T+1
    if they are different languages, count the instance as post-T codeswitch
```

Figure 2: Analysis algorithm for words

Run the above with progressively looser constraints on T: homophonic nouns, plurals of homophonic nouns, proper nouns, adjectives, verbs in -[i]o, interactional markers

Consider extending the window from 1 word before and after to 2, 3, etc? If we wanted to see how far away the switch was from the cognate in the clause as well, that would be extra information and new to our study. Eva Eppler has a Distance Hypothesis that refers to this.

You could also look at cognate density in the area of a codeswitch, ie how many cognates occur in the pre and post ?5 utterances, and see if that is significant.

Mirjam: At the word level it would for example be the number of words:

1. following a cognate + switched
2. following a cognate + not switched
3. not following a cognate + switched
4. not following a cognate + not switched
5. preceding a cognate + switched
6. preceding a cognate + not switched
7. not preceding a cognate + switched
8. not preceding a cognate + not switched

4 Tagging

Initial work has assumed that anything tagged @s:cym&eng is a cognate. Maybe we should just ignore the Siarad tagging completely and rely solely on selection based on the lists of cognates? The existing tagging is useful for proper nouns but we will only be bringing the proper noun cohort in at a later stage:

So maybe for the first two cohorts we should actually exclude them?

5 Additional questions to investigate

Mirjam: The above focusses on a replication of my earlier work. That's a good start, and we'll definitely have to report that. But of course, we also want to add something new. So we need to think about the questions we raised in our proposal for the British Academy (or any other interesting questions).

[So perhaps we only need to do a small sample based on her approach, so that we can compare the validity of the new stuff with her old stuff, and major on the new stuff?]

In order to decide what new questions we want to tackle, we need to think about what the corpus has to offer. We should maximally benefit from the

information that's already present in the corpus. Therefore, this is mainly something you will have to work out, with yourknowledge of the corpus.

Some things we might be able to address:

1. Can we look at false friends and compare them to cognates - have FFs been coded in the corpus? [Do we have a list of these?]

2. Can we find out whether grammatical function affects triggered CS? Are cognates more likely to trigger a switch when they are subject than when they are object, for example? Or are subjects more likely to 'undergo' triggered CS than objects?

3. And similarly for word classes: are some word classes more likely to cause or undergo triggered CS than others? [This and teh previous point amount to the same thing, since we only have POS tagging at this point, and the POS tag would have to be used as a proxy for syntactic role.]

For the definitions we want to use, one way to go is to stick to the definitions I've been using, but the other option is to go with what you have in the corpus. For the cognates, I would like to stick to the codes that are already there.

6 Scoping the additional questions

It would also be a good idea to write out some "plan" for each issue you want to look at:

- hypothesis – what we want to prove
- input – what aspects of the Siarad data we will look at
- method – what we plan to do to the unfortunate data
- outputs – what we need to prove the hypothesis

We should then do some small runs on (say) 10 files, to ensure we are getting useable data. If you want to analyse these statistically, perhaps we could use an R script to do this automatically, to lower the cost of subsetting the data.