A-Not-A in a TPG

Marcus Smith

Last Modified: June 16, 2004

0 Introduction

Chinese A-Not-A constructions have been analyzed as involving a operation that copies syntactic structures of arbitrary complexity (Huang, 1991). This paper demonstrates that Chinese A-Not-A constructions can be handled without any such copying process by the Tupled Pregroup Grammar formalism of Stabler (2004).

The first section will give an overview of the construction. Following that, a grammar for “Anglicized Mandarin” will be presented. This mini-grammar will capture the essential properties of Mandarin A-Not-A questions, including some that are not strictly relevant to the question of copying, just to highlight the general applicability of the approach. Some difficulties and how they correspond to actual Mandarin will be discussed in section 4.

1 Overview of A-Not-A

Polar questions in Mandarin that are not biased towards one answer or another are expressed through apparent “copying” of the verb or (optionally) the verb phrase, and conjoining the copies with the negative adverbial 男 ‘not’.1 Huang (1991) cites the Mandarin sentences in (1) as examples.

(1) a. ta xihuan zheben shu bu xihuan zheben shu?
   he like this book not like this book
   ‘Does he like this book or doesn’t [he] like this book?’

b. ta xihuan bu xihuan zheben shu?
   he like not like this book
   ‘Does he like or doesn’t [he] like this book?’

c. ta xihuan zheben shu bu xihuan?
   he like this book not like
   ‘Does he like this book or doesn’t [he] like [it]?’

1Mandarin also has a negator mei which has a different distribution from 男. Many Chinese languages, such as Cantonese, use cognates of mei in their A-not-A constructions. (Law, 2001; Cole and Lee, 1997).
d. ta xi-bu-xihuan zheben shu?
   he like-not-like this book
   ‘Does he like or not like this book?’

The first sentence is a complete copy of the verb and its object. The next
two are partial copies, where the verb is duplicated, but the object only occurs
once. Note that the object can be adjoined to either copy of the verb. The
last sentence is the most minimal, with the first copy consisting of only the first
syllable of the verb. Crucially, the negative must appear between both copies,
and the two copies must match within the limits of variation seen above.

Beyond main verbs and the complements, it is possible to copy auxiliary
verbs and manner adverbs as well (Chen and Hong, 1998). The latter are less
acceptable for many speakers, however (Ying Lin, p.c.).

(2) ta hui bu hui mai nadong fangzi?
    he will not will buy this-CL house
    ‘Will he [or] won’t [he] buy this house?’

(3) ta manmande pao bu manmande pao?
    he slowly run not slowly run
    ‘Does he [or] doesn’t [he] run slowly?’

A-Not-A questions are not compatible with certain other elements. In particu-
lar, it is not possible to use this question formation strategy when the du-
plicated elements would be preceded by a quantificational element (Wu, 1997).
(4a), for example, is ungrammatical because of the quantified subject. This can
be avoided by inserting the copula shi ‘be’, and copying it. The rest of the
sentence is not copied in such structures.

(4) a. *you ren pao bu pao?
    someone run not run
    ‘Does someone run or not?’
b. shi-bu-shi you ren pao?
    be-not-be someone run
    ‘It is the case or not that everyone runs?’

I follow, Huang (1991), Cole and Lee (1997), and Law (2001) in using a silent
question operator that moves the CP region. Quantified phrases have their own
operators, which block the question operator as a minimality violation. See Wu
(1997) for an alternative analysis based on the semantics of the construction.

2 Tupled Pregroup Grammars

The following definitions are taken from Stabler (2004). A Tupled Pregroup
Grammar (TPG) has a set of simple types $P$, and for each simple type $a$, there

\footnote{It is also possible to have a sentence like (1c except that the last verb is missing: ta xihuan zheben shu bu ‘Does he like this book or not?’. In these contexts, bu shifts from falling tone to neutral tone (Ying Lin, p.c.). I ignore this construction below.}

2
is a left adjoint \( a^! \) and a right adjoint \( a^* \). A finite sequence of simple types and adjoints is a “type”. A partial order can be defined on the simple types, such that if \( a \leq b \), then \( a \) is a more specific type of \( b \). Strings in the language are taken from a finite alphabet \( \Sigma \) plus the empty string \( \epsilon \). Expressions in the grammar are tuples of string-type pairs: \((\Sigma^* \times \text{Type})^k\). Expressions will be written with types over strings \( \left( \begin{array}{c} t_0 & \ldots & t_i \\ s_0 & \ldots & s_i \end{array} \right) \) or types before strings \( t_0, \ldots, t_i : s_0, \ldots, s_i \), as convenient.

An expression is “saturated” iff no adjoint types occur in any coordinate of it. And an expression is “proper” iff the following conditions hold:

1. Every type in it has exactly one atom,
2. No two types have the same atom,
3. \( \text{GCON} \) does not apply to it, or any result of applying \( \text{Move} \) to it.

\( \text{GCON} \) is a contraction operation which applies to a coordinate in an expression. If the type has a sequence of simple type and an appropriate adjoint, both can be removed. The following hold if \( a \leq b \).

\[
\begin{pmatrix}
\ldots & xab^ry \\
\ldots & s \\
\ldots & x^by^ay
\end{pmatrix} \rightarrow \begin{pmatrix}
\ldots & xy \\
\ldots & s \\
\ldots & xy \\
\end{pmatrix}
\]

If two expressions are proper, and at least one of them is saturated, then they can be merged by the following rule.

\[
\begin{pmatrix}
t_1 & \ldots & t_i \\
s_1 & \ldots & s_i
\end{pmatrix} \cdot \begin{pmatrix}
t_{i+1} & \ldots & t_k \\
s_{i+1} & \ldots & s_k
\end{pmatrix} = \begin{pmatrix}
t_1 & \ldots & t_k \\
s_1 & \ldots & s_k
\end{pmatrix}
\]

“\( \text{Move} \)” applies to a single expression with more than one coordinate. It takes two coordinates, removes them from the expression, unifies them into a single expression, then merges the result back with the remnants of the original expression. \( \text{Move} \) applies to a pair of coordinates only if both are proper and at least one of them is saturated.

\[
\begin{pmatrix}
t_1 & \ldots & t_k \\
s_1 & \ldots & s_k
\end{pmatrix} \rightarrow \begin{pmatrix}
t_{i,j} & \ldots & t_k \\
s_{i,j} & \ldots & s_k
\end{pmatrix} \cdot \begin{pmatrix}
t_1 & \ldots & t_k \\
s_1 & \ldots & s_k
\end{pmatrix}^{-i,j}
\]

3 A TPG for A-Not-A

The A-Not-A constructions seen above can be modeled simply with a TPG. The following is a TPG for Anglicized Mandarin (AM) – a relexification of the sentences in section 1 above. The sentences have been anglicized to avoid certain irrelevant complications, such as the structure of nominal classifiers. First, the basic expressions needed to derive declarative statements will be given and their use illustrated. Following that, expressions necessary for A-Not-A constructions will be added.
AM has the simple types in the table below.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Auxiliary</td>
<td>A₀</td>
</tr>
<tr>
<td>α</td>
<td>Adverb</td>
<td>α₀</td>
</tr>
<tr>
<td>D</td>
<td>Determiner/Pronoun</td>
<td>D₀</td>
</tr>
<tr>
<td>N</td>
<td>Noun</td>
<td>N₀</td>
</tr>
<tr>
<td>V</td>
<td>Verb</td>
<td>V₀</td>
</tr>
<tr>
<td>O</td>
<td>Operator</td>
<td>O₀</td>
</tr>
<tr>
<td>G</td>
<td>Negative</td>
<td>S</td>
</tr>
</tbody>
</table>

3.1 Declarative Sentences

The determiners in the language are given below. The first is a simple pronoun. The second is a demonstrative that selects a nominal complement. For simplicity, the quantifier *some* is treated as a determiner with an associated operator.

\[
\left( D \right) \left( D N^l \right) \left( D N^l \right) \left( O \right)
\]

There are three nouns.

\[
\left( N \right) \left( N \right) \left( N \right)
\]

The language has four verbs: the intransitive *run*, and the transitive verbs *like* and *purchase*. The latter two select expressions headed by a determiner. The fourth verb is the copula *be*, which selects a sentential object. Note that by taking this approach, I am ignoring issues regarding the relationship between the subject and verb. While this is a non-trivial simplification, it does not affect the ability of TPGs to handle the A-Not-A data.

\[
\left( V \right) \left( V D^l \right) \left( V D^l \right) \left( C S^l \right)
\]

The only adverb is *slowly*, which attaches to the left of a verb. For reasons that will become clear, I follow the suggestion of Cinque (1999) that each adverb heads its own projection rather than the classical adjunction model.

\[
\left( \alpha V^l \right)
\]

There is one auxiliary verb, which combines with a verbal expression to the right.

\[
\left( A V^l \right)
\]

There are multiple types of negation, depending on what size of constituent is being negated. Here, there are three possibilities: negation of the verb, adverb, or auxiliary.

\[
\left( G V^l \right) \left( G a^l \right) \left( G A^l \right)
\]

4
Finally, we have the expressions necessary to form a complete sentence. Since polar questions with *ma* in Mandarin place the question particle at the end of the sentence, the S-forming expressions, though null, take all their arguments to the left. Unlike English, Mandarin does not require a visible subject for all sentences, so some S-forming expressions in AM require an expression of type D and others do not. Questions and sentences with quantified subjects require operator movement, so this is factored in as well. Finally, the complement of a sentence can be an auxiliary phrase, negative phrase, adverb phrase, or verb phrase, so each of these need to be accommodated by distinct expressions.

\[
\begin{pmatrix}
G^* \cdot D^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
G^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
G^* \cdot D^* O^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
G^* O^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
A^* \cdot D^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
A^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
A^* \cdot D^* O^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
A^* O^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
\alpha^* \cdot D^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
\alpha^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
\alpha^* \cdot D^* O^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
\alpha^* O^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
V^* \cdot D^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
V^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
V^* \cdot D^* O^* S \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
V^* O^* S \\
\epsilon
\end{pmatrix}
\]

These lexical expressions can generate the declarative versions of all the sentences in section 1.

### 3.2 A-Not-A Sentences

Adding A-Not-A constructions requires the addition of new expressions homophonous with negation, and doubled versions of many lexical items.

Based on comparative and historical work by Cole and Lee (1997) and Schaffar (2002), it likely that the negator *bu* used in Mandarin A-Not-A sentences is the result of conflating three separate elements over time: a question particle (*keh* is Classical Mandarin, *ka* is modern Singapore Teochow), a disjunctive particles (*hisashi* in modern Mandarin), and negation. Because of this the negative in AM selects for both copies (the function of the disjunctive particle) and has a question operator (the function of the question particle), while having the same form as the negative. Since questions can be formed by copying auxiliaries, adverbs, and verbs, three forms of question negator are necessary.

\[
\begin{pmatrix}
A_2^* \cdot G^l \cdot O \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
V_2^* \cdot G^l \cdot O \\
\epsilon
\end{pmatrix}
\begin{pmatrix}
\alpha_2^* \cdot G^l \cdot O \\
\epsilon
\end{pmatrix}
\]

Auxiliaries, Verbs, Adverbs, Determiners, and Nouns (and others not discussed here) occur in doubled forms. For the final three, the description of the doubled expressions is simple. Each coordinate in the basic expression is a pair of coordinates in the doubled version. The string portion of each coordinate
pair is identical. The types for each coordinate pair are similar, except that in one, each simple type and adjoint have an index 2.

\[
\begin{pmatrix}
D & D_2 \\
\text{he} & \text{he}
\end{pmatrix}
\begin{pmatrix}
DN^l & D_2N_2^l \\
\text{this} & \text{this}
\end{pmatrix}
\begin{pmatrix}
DN^l & D_2N_2^l & O_2 \\
\text{some} & \text{some} & \epsilon
\end{pmatrix}
\begin{pmatrix}
N & N_2 \\
\text{book} & \text{book}
\end{pmatrix}
\begin{pmatrix}
N & N_2 \\
\text{house} & \text{house}
\end{pmatrix}
\begin{pmatrix}
N & N_2 \\
\text{person} & \text{person}
\end{pmatrix}
\begin{pmatrix}
\alpha V^l & \alpha_2 V_2^l \\
\text{slowly} & \text{slowly}
\end{pmatrix}
\]

Verbs and auxiliaries are essentially the same, but slightly more complex. Complete copying of the verb and complement is achieved in precisely the same way: paired coordinates with matching strings and almost matching types. The most natural A-Not-A questions do not have this full copying, however, so other patterns are needed. One pattern is to have only the simple type of a coordinate matching, with all adjectives removed. This can hold for either the V or V_2 and the A or A_2. For polysyllabic words, there is a doubled expression where only the first syllable of the V_2 or A_2 is matched; in these expressions the V_2 or A_2 has no adjectives. The doubled verbs and auxiliaries are:

\[
\begin{pmatrix}
V & V_2 \\
\text{run} & \text{run}
\end{pmatrix}
\begin{pmatrix}
VD^l & V_2D_2^l \\
\text{like} & \text{like}
\end{pmatrix}
\begin{pmatrix}
VD^l & V_2 \\
\text{like} & \text{like}
\end{pmatrix}
\begin{pmatrix}
VD^l & V_2 \\
purchase & purchase
\end{pmatrix}
\begin{pmatrix}
VD^l & V_2 \\
purchase & purchase
\end{pmatrix}
\begin{pmatrix}
VD^l & V_2 \\
purchase & pur
\end{pmatrix}
\begin{pmatrix}
CS^l & C_2S_2^l \\
\text{be} & \text{be}
\end{pmatrix}
\begin{pmatrix}
CS^l & C_2S_2^l \\
\text{be} & \text{be}
\end{pmatrix}
\begin{pmatrix}
CS^l & C_2 \\
\text{be} & \text{be}
\end{pmatrix}
\begin{pmatrix}
AV^l & A_2V_2^l \\
\text{will} & \text{will}
\end{pmatrix}
\begin{pmatrix}
AV^l & A_2 \\
\text{will} & \text{will}
\end{pmatrix}
\begin{pmatrix}
AV^l & A_2 \\
\text{will} & \text{will}
\end{pmatrix}
\]

Two of the expressions for be are useless since there are no S_2’s in AM. Thus, only questions formed from be can only occur in a form where the embedded sentence follows both copies. This appears to match the empirical data on Mandarin. The expressions are included here because they do not cause any harm.

Based on these additions, all the questions seen in section 1 can be derived. Take, for example, (8), the relexification of (1a).

(8) You like this book not like this book.

All doubled words must match – it is not possible to change the noun or verb and still derive the sentence (9). This effect is the result of a “conspiracy” in the types. *not* must combine with elements of the same type. The only way to
get this is to use a doubled expression. This rules out sentences like \((9a)\). Each verb is specified in the lexicon as to what kind of expressions it combines with. It would not be possible to have two unmatched nouns, as in \((9b)\), because in order for a sentence with duplicated verbs to have two objects, one copy of the object must be a \(D_{e}\). But this is only possible if the object has been doubled, forcing the objects to match. The end result is that the “copies” can only vary in ways stipulated in the lexicon.

\[(9)\]
\begin{align*}
&\text{a. * You like this book not purchase this book.} \\
&\text{b. * You like this book not like this house.}
\end{align*}

Consider again AM sentences \((10a)\) and \((10b)\), relexifications of \((4a)\) and \((4b)\), respectively. Because both the subject and the negative have silent operators but there is only one position for an operator in the sentence, the sentence in \((10a)\) cannot be derived. In \((10b)\), the operator associated with the subject is used in the formation of an embedded sentence. This sentence is selected by a doubled version of \textit{be}, which can successfully be used with in an A-Not-A question. Thus, AM is capable of modeling actual Mandarin patterns.

\[(10)\]
\begin{align*}
&\text{a. * Some person run not run?} \\
&\text{b. Be not be some person run?}
\end{align*}

4 Redundancies, Ambiguities, and Reality

The lexicon given for AM is highly redundant. Some words are listed multiple times with no purpose other than to control exactly what category it combines with. It would seem more elegant to use partial orders to factor some of this out. For example, by defining the partial order \(V\alpha AG\), it would be possible to have just two expression for forming a sentence, one for sentences with quantifiers, the other for those without. The expression would formally select \(G\), but would be able to combine with \(A\)'s, \(V\)'s, and \(\alpha\)'s due to the partial order. However, it turns out that doing so would weaken the requirement on strict copying, as will be discussed shortly.

The excesses seen here could be remedied simply by introducing lexical redundancy rules. Each expressions could be stipulated once, and the rules would produce the additional forms. Such rules would also permit the verbs and other duplicatable words to be entered once.\(^3\)

Reducing redundancy through partial orders undermines the means by which the copies are forced to be identical. The word \textit{not} selects two expressions differing only in that one carries a 2 subscript. If, say, \(V\) is specified as a subtype of \(A\), then it would be possible for \textit{not} to combine with a auxiliary expression on the right, but a verbal expression on the left, as long as the auxiliary expression

\(^3\)This is how the grammar was produced for machine testing. I wrote the basic entries, and a perl script produced all predictable entries.
contains a copy of the verb. Similar considerations prevented treatment of adverbs as adjuncts of type \( VV' \).

11. \( V \) like this music \( \) not \( A \) will like this music \( . \)

12. \( V \) run \( \) not \( V \) slowly run \( . \)

The point can be generalized to give a limitation on what kind of copying structures are possible. Anytime the copied structures include a type \( a \), and there is a type \( b \), such that \( a \leq b \), exact duplication cannot be guaranteed. The reason is that the structure building operations available only have access to the types of the expressions. Thus, if there is any ambiguity in which expression attaches at a particular spot, all options are possible. Because of this, several types of syntactic structures cannot be accurately copied with the current approach.

Mandarin allows serial verb constructions. (Po-Ching and Rimmington, 1997) Unless each verb in such a sequence has a slightly different type, the approach used here will not be able to guarantee exact duplication. Note that verbs can appear at the beginning or the end of serial constructions (e.g., \( gu \) ‘go’ below), so it may not be possible to build the proper structures into the types.

13. a. Ta xia le ke hui jia qu le.
   he finish ASP class return home go P
   ‘He finished class and went home.’

   b. Wo qu shangdian mai dongxi.
   I go shop buy thing
   ‘I am going to the shops to do some shopping.’

Embedded sentences are also problematic. If the \( S \) types were added into the grammar, it would be impossible to force the embedded sentences to be exact copies. Nothing would be able to force the arguments of the embedded clause to map to the same grammatical role; that is, swapping subject and object roles in one copy would not lead to an ungrammatical structure. Even if grammatical role were built into the types, the problem related to auxiliary and main verb would be replicated. Not would be able to combine the embedded verb phrase with the matrix verb phrase. If the verb say were added to AM, then the following would be permissible.

14. \( V \) like this book \( \) not \( V \) say he like this book \( . \)

There is a simple and realistic solution to these problems: to not do full copying of complements. AM allows for sentences where the auxiliary verb and its complement are all copied (15), but this does not happen in Mandarin. A-Not-A questions formed from auxiliaries only copy the auxiliary, not the rest of the sentence. The AM equivalent would be (16).

15. He will buy this house not will buy this house.

16. He will not will buy this house.
Mandarin does not allow copying of embedded sentences or serial verbs, so exactly the constructions that are causing difficulty for the current approach are excluded for the natural language being mimicked. An accurate model of Mandarin, then, should only allow the duplicate strategy wherein the $V$ (or $A$) coordinate selects a complement, but the $V_2$ (or $A_2$) does not. This would hold of any verb or auxiliary that selected a verb or sentence.

5 Conclusion

The Chinese A-Not-A constructions can be modeled with a fair degree of accuracy as a TPG. The nature of the formalism causes difficulties for certain types of structures, but luckily the natural language being modeled does not allow those constructions anyways. This is a welcome result, in a sense. The disallowed constructions are difficult to derive using just the TPG operations, but a copying approach would not have any such problems. If syntactic structures were simply copied across, then one wonders why not all potentially copiable structures are acceptable. It is tempting to suggest that the reason some structures do not copy is because allowing them would permit additional, anomalous syntactic structures as a side-effect. Such a conclusion would be unwarranted, given the information available, but it poses interesting directions for further research.

References


\footnote{Mandarin appears to be very liberal in its copying constructions. Most other Chinese languages only permit the copying of the head verb/auxiliary itself, never its complement. (Cole and Lee, 1997; Law, 2001; Schaffar, 2002)}
