Table of Contents

Phenomena specified by the LinGO Grammar Matrix customization system

Word Order

The basic word order in Mandarin Chinese is subject-verb-object (SVO), with determiners and auxiliary verbs precede nouns and complements respectively. These phenomena are specified in the `Word Order` section of the customization system.

Number, Person, Gender

Not specified.

Case

Not specified.

Adnominal Possession

Not specified.

Direct-inverse

Not specified.

Tense, Aspect and Mood

Verb tenses are not expressed in Mandarin Chinese. Instead several particles are used to express verbal aspect, as given below:

- Perfective aspect "了" (e.g. "他给了她。")
- Imperfective aspect "在" (e.g. "他在唱歌。")
- Experiential aspect "过" (e.g. "他给过她。")

Evidentials

Not specified.

Other Features

Although Mandarin Chinese does not distinguish between finite and non-finite verb form, the `Add finite/non-finite FORM feature distinction.` option is still selected as it is a mandatory requirement when auxiliary verbs were specified in the `Word Order` section.

Sentential Negation

Negation construction in Mandarin Chinese is specified as a `simple` morphosyntactic exponence, in which a negative auxiliary verb "不" is introduced.

Coordination

Two coordination strategies for the Mandarin Chinese Grammar are specified:

- Monosyndeton (e.g. "我 你 和 她")
- Polysyndeton (e.g. "我 和 你 和 她")

Matrix Yes/No Questions

A question in the Mandarin Chinese Grammar is formed by introducing a sentence final particle "吗".

Information Structure

Not specified.

Argument Optionality

In Mandarin Chinese, subject dropping can occur with any verb in any context (e.g. "狗追猫。" and "追猫！" are both sensible sentences) while object dropping can only occur with certain lexical entries (e.g. dropping "猫" in "他有猫。" is ok but not "猫" in "狗追猫。").

Nominalized Clauses

Not specified.

Clausal Complements

Two types of clausal complements are specified in the Mandarin Chinese Grammar:

- Clausal complements appearing in the same position as regular noun complements without a complementizer (e.g. "她觉得他会唱歌。" with "他会唱歌" as a clausal complement of "觉得").
- Clausal complements appearing in the same position as regular noun complements with a complementizer "是否" (e.g. "她问他是否会唱歌。" with "会唱歌" as a clausal complement of "问").

Clausal Modifiers

Not specified.

Morphology

Not applicable.

Phenomena specified by modifying .tdl files

Complement of Auxiliary Verb

The auxiliary verb "要" is assigned to be a subclass of `common-aux-lex` by the grammar customization system, which permits usage of double auxiliary verbs. Hence, sentences such as "他不要唱歌。" and "他要不唱歌。" will produce parse results. However, the latter parse result is invalid because "他要不唱歌。" is a grammatically incorrect sentence. To rectify this issue, a new lexical rule `yao-aux-lex` is added to make sure that the complement of "要" is not an auxiliary verb:

```
yao-aux-lex := common-aux-lex &
  [ SYNSEM.LOCAL.CAT.VAL.COMPS.FIRST.LOCAL.CAT.HEAD.AUX -].
```

Subsequently, the lexical entry of the auxiliary verb "要" is modified as follow:

```

要_aux := yao-aux-lex &
  [ STEM < "要" >,
    SYNSEM.LKEYS.KEYREL.PRED "_want_v_rel" ].
```

Demonstratives

Determiners such as "那个"，"一只" can either be used as a quantifier for other nouns to form a noun phrase (e.g. "那个人", "一只猫") or as a standalone demonstrative pronoun. If they are used as demonstrative pronouns, they contribute multiple predicates to the semantics, which can be specified via the following lexical rule:

```
noun+det-lex-item := norm-hook-lex-item & non-mod-lex-item &
    [SYNSEM [LOCAL [CAT [ HEAD noun,
                          VAL [ SPR < >, COMPS < >,
                                SUBJ < >, SPEC < > ]],
                    CONT [RELS <! relation &
                            [LBL #nh, ARG0 #s ],
                            quant-relation & #det &
                            [ARG0 #s, RSTR #h ]!>,
                         HCONS <! qeq & [ HARG #h,
                                          LARG #nh ] !> ]],
            LKEYS [ KEYREL relation,
                    ALTKEYREL #det ]]].

n+det-lex := noun+det-lex-item.
```

Some example lexical entries of demonstrative pronouns：

```
那只_n := n+det-lex &
  [ STEM < "那只" >,
    SYNSEM.LKEYS.KEYREL.PRED "_animal_n_rel",
    SYNSEM.LKEYS.ALTKEYREL.PRED "_that_q_rel" ].

一只_n := n+det-lex &
  [ STEM < "一只" >,
    SYNSEM.LKEYS.KEYREL.PRED "_animal_n_rel",
    SYNSEM.LKEYS.ALTKEYREL.PRED "_a_q_rel" ].

那个_n := n+det-lex &
  [ STEM < "那个" >,
    SYNSEM.LKEYS.KEYREL.PRED "_thing_n_rel",
    SYNSEM.LKEYS.ALTKEYREL.PRED "_that_q_rel" ].
```

Complements of Ditransitive Verb

The complements of a ditransitive verb consist of a noun phrase and an embedded clause. For example, given the sentence "他问她是否会唱歌。", "问" is the ditransitive verb, "她" and "是否会唱歌" are the noun phrase and embedded clause respectively. This phenomenon is specified via the following lexical rules:

```
ditr-clausal-verb-lex := main-verb-lex &
  [ SYNSEM.LOCAL.CAT.VAL.COMPS < #obj, #clause >,
    ARG-ST < [ LOCAL.CAT.HEAD noun ],
             #obj &
             [ LOCAL.CAT [ HEAD noun,
                           VAL.SPR < > ] ],
             #clause &
             [ LOCAL.CAT.VAL [ SPR < >,
                               COMPS < >,
                               SUBJ < > ] ] > ].

ditr_comp-verb-lex := ditr-clausal-verb-lex & clausal-third-arg-ditrans-lex-item &
```

```
  [ SYNSEM.LOCAL.CAT.VAL.COMPS < [], [ LOCAL.CAT.HEAD comp ] > ].
```

Follow by the lexical entries of ditransitive "问":

```
问_ditr := ditr_comp-verb-lex &
  [ STEM < "问" >,
    SYNSEM.LKEYS.KEYREL.PRED "_ask_v_rel" ].
```

Adverb

Some adjectives can be combined with the clitic '地' to form an adverb of manner. For example, given the sentence "她高兴地唱歌。", the adjective "高兴" is combined with "地" to form an adverb which modifies the verb "唱歌". This phenomenon is specified via the following lexical rules:

```
adj-to-adv-lex := no-icons-lex-item & raise-sem-lex-item &
                  trans-first-arg-raising-lex-item-2 &
                  intersective-mod-lex &
    [ SYNSEM.LOCAL.CAT.HEAD adp,
      SYNSEM.LOCAL.CAT.VAL.COMPS <[LOCAL.CAT.HEAD verb]>,
      SYNSEM.LOCAL.CAT.HEAD.MOD <[LOCAL.CAT.HEAD adj]> ] .
```

Follow by the lexical entries of "地":

```
地_adp := adj-to-adv-lex &
  [ STEM < "地" > ].
```

Aspect Marker

Aspect markers such as "了" and "过" generally follow after a verb (e.g. "他给了她。", "他当过兵。"）. To model this phenomenon, a new rule called `head-marker-phrase` is introduced:

```
head-marker-phrase := basic-head-marker-phrase & marker-final-phrase & head-initial.
```

A `head-marker-phrase` is a phrase headed by a verb at the initial position followed by an aspect marker. The feature structure of `basic-head-marker-phrase` is given by:

```
basic-head-marker-phrase :=  head-valence-phrase & head-compositional & basic-binary-
headed-phrase &
  [ SYNSEM phr-synsem & [ LOCAL.CAT [ VAL [ SUBJ #subj,
                                            COMPS #comps,
                                             SPR #spr,
                                             SPEC #spec ],
                                       POSTHEAD #ph ] ],
    HEAD-DTR.SYNSEM #synsem & [ LOCAL [ CAT [ VAL [ SUBJ #subj,
                                                      COMPS #comps,
                                                      SPR #spr,
                                                      SPEC #spec ],
                                               POSTHEAD #ph ] ] ],
    NON-HEAD-DTR [ SYNSEM [ LOCAL.CAT [  HEAD marker,
                                         VAL.COMPS < #synsem > ] ] ],
    C-CONT [ RELS <! !>,
             HCONS <! !>,
             ICONS <! !> ] ].
```

The marker type and the corresponding lexical rule which form the non-daughter part of `head-marker-phrase` are defined as followed:

```
aspect-marker := raise-sem-lex-item & no-icons-lex-item &
                  norm-zero-arg &
 [ SYNSEM [ LOCAL [ CAT [ HEAD marker & [ MOD <> ],
                            VAL [ COMPS < lex-synsem &
                                          [ LOCAL [ CAT [ HEAD verb & [ AUX - ],
                                                         VAL.SUBJ < [] > ] ] ] >,
                                  SUBJ < >,
                                  SPEC < >,
                                  SPR < > ],
                            POSTHEAD + ] ] ] ].

aspect-lex-item := aspect-marker &
 [ SYNSEM.LOCAL.CAT.VAL [ COMPS < [ LOCAL [CONT.HOOK.INDEX.E.ASPECT perfective,
                                           CAT.HEAD verb ] ] > ] ].
```

Parse results

True Positive

```
delphin select 'i-id i-input where i-wf = 1 and readings > 0' trees/testsuite.01/
```

No.	Sentence	Phenomena
1.	那只 狗 追 一只 猫	Word order
2.	一只 猫 追 一只 狗	Word order
3.	他 在 唱歌	Word order

No.	Sentence	Phenomena
4.	他 给 了 她	Word order
5.	他 给 了 她 一只 猫	Word order
6.	他 要 唱歌	Word order
7.	他 应该 会 唱歌	Word order
8.	我 会	Pronouns
9.	我 给 了 他 一只 猫	Pronouns
10.	他 给 了 我 一只 猫	Pronouns
11.	他 给 了 那个 人 一只 猫	Pronouns
12.	那里 有 一只 猫	Case
13.	那里 有 猫	Case
14.	那里 有	Case
15.	那只 猫 在 唱歌	Determiners
16.	猫 在 唱歌	Determiners
17.	小明 在 唱歌	Determiners, Tense Aspect Mood
18.	他 给 了 我 一只 猫	Tense Aspect Mood
19.	他 给 过 我 一只 猫	Tense Aspect Mood
20.	他 不 要 唱歌	Negation
21.	他 没 有 猫	Negation
22.	他 有 猫	Argument optionality
23.	他 有	Argument optionality
24.	追 猫	Argument optionality
25.	这只 猫	Cognitive status
26.	那只 猫	Cognitive status

No.	Sentence	Phenomena
27.	一只 猫	Cognitive status
28.	他 会 唱歌 吗	Matrix yes-no questions
29.	他 不 会 唱歌 吗	Matrix yes-no questions, negation
30.	我 和 他 追 一只 猫	Coordination
31.	我 小明 和 他 追 一只 猫	Coordination
32.	我 和 小明 和 他 追 一只 猫	Coordination
33.	她 高兴 地 唱歌	Adverbs
34.	她 觉得 他 不 会 唱歌	Embedded declaratives
35.	她 问 他 是否 会 唱歌	Embedded questions
36.	那只 猫 很 可爱	Non-Verbal Predicates
37.	她 要 一只 白 猫	Adjectives
38.	她 大概 知道	Adverbs

True Negative

```
delphin select 'i-id i-input where i-wf = 0 and readings = 0' trees/testsuite.01/
```

No.	Sentence	Phenomena	Remarks
1.	一只 狗 一只 猫 追	Word order	SOV word order
2.	一只 猫 那只 狗 追	Word order	OSV word order
3.	追 那只 狗 一只 猫	Word order	VSO word order
4.	追 一只 猫 那只 狗	Word order	VOS word order
5.	他 在 唱歌 她	Word order	Object in intransitive verb
6.	他 唱歌 要	Word order	Auxiliary verb after verb

No.	Sentence	Phenomena	Remarks
7.	他 唱歌 应该 会	Word order	Auxiliary verb after verb
8.	他 应该 唱歌 会	Word order	Auxiliary verb after verb
9.	他 会 唱歌 应该	Word order	Auxiliary verb after verb
10.	他 给 了 那个 我 一只 猫	Pronouns	Determiner before pronoun
11.	猫 那只 在 唱歌	Determiners	Determiner after noun
12.	猫 在 那只 唱歌	Determiners	Determiner after noun
13.	猫 在 唱歌 那只	Determiners	Determiner after noun
14.	那个 小明 在 唱歌	Determiners	Determiner before proper noun
15.	他 给 我 了 一只 猫	Tense Aspect	Perfective aspect after pronoun
16.	他 给 我 过 一只 猫	Tense Aspect	Experiential aspect after pronoun
17.	在 唱歌 小明	Tense Aspect	Subject after imperfective aspect
18.	他 要 不 唱歌	Negation	Negative auxiliary verb after verb
19.	他 有 没 猫	Negation	Negative auxiliary verb after verb
20.	狗 追	Argument	Invalid object dropping
21.	猫 那只	Cognitive	Demonstrative after noun
22.	猫 一只	Cognitive	Demonstrative after noun
23.	他 会 吗 唱歌	Matrix yes-no	Sentence final particle before words
24.	他 吗 会 唱歌	Matrix yes-no	Sentence final particle before words
25.	吗 他 会 唱歌	Matrix yes-no	Sentence final particle before words
26.	我 小明 他 追 一只 猫	Coordination	Missing coordinator
27.	和 我 小明 和 他 追 一只 猫	Coordination	Sentence initial coordinator
28.	和 我 和 小明 和 他 追 一只 猫	Coordination	Sentence initial coordinator
29.	她 要 一只 猫 可爱 的	Adjectives	Adjective after noun

No.	Sentence	Phenomena	Remarks
30.	她 地 高兴 唱歌	Adverbs	Adverb before adjective
31.	她 觉得 他 是否 不 会 唱歌	Embedded	Invalid complementizer
32.	她 问 他 会 唱歌	Embedded	Missing complementizer or sentence final
33.	那只 猫 可爱	Non-Verbal	Missing modifier
34.	她 要 一只 猫 白	Adjectives	Adjective after noun
35.	她 知道 大概	Adjectives	Adverb after verb

False Positive

```
delphin select 'i-id i-input where i-wf = 0 and readings > 0' trees/testsuite.01/
```

No false positive

False Negative

```
delphin select 'i-id i-input where i-wf = 1 and readings = 0' trees/testsuite.01/
```

No.	Sentence	Phenomena	Remarks
1.	追	Argument optionality	Fragment
2.	她 要 一只 可爱 的 猫	Adjectives	Relative marker

Trigger rules

Trigger rules (defined in `trigger.mtr`) controls the generation of lexical entries with empty semantics. An example of lexical entries with empty semantics is the copula "很" (different than the adverbial "很" which means "very" in English), that is used to link a subject to an adjective (e.g. "猫很可爱。"). A trigger rule is defined for the copula "很" so that its lexical entry is added to the generator chart whenever an adjective predicate is encountered during generation or translation:

```
很_gr := generator_rule &
  [ CONTEXT.RELS <! [ PRED "~_a_" ] !>,
    FLAGS.TRIGGER "很" ].
```

Variable property mapping

As the aspect markers such as "在", "了" are semantically empty, they would be added to the generator chart according to the trigger rules created by the matrix customization system, i.e. whenever there is a specified/underspecified `ARG0.E.ASPECT` in the minimal recursion semantics (MRS). In the case of underspecified `ARG0.E.ASPECT`, both perfective and imperfective aspect markers will be added and produced many unintended sentences. To constrained this over-generative behavior, the following mappings is added to the `semi.vpm` file to map underspecified aspects to a default aspect type `no_aspect`:

```
E.ASPECT : E.ASPECT
  perfective <> perfective
  imperfective <> imperfective
  no_aspect << *
```

Limitations

A list of known issues/limitations of the grammar is as follow:

1. Incorrect parse results for manner adverb "地" (e.g. in the parse tree of the sentence "他高兴地唱歌", "地" is incorrectly attached to the verb "唱歌" instead of the adjective "高兴")
2. Recursive auxiliaries (e.g. ungrammatical sentence with multiple auxiliary verbs like "她可以可以可以吃。" will produce parse results)

3. Demonstrative "那里" is hard-coded as a proper noun to parse sentence like "那里有人。"
4. No distinction between classifiers such as "只" and "个"
5. No distinction between negative particles such as "没" and "不"

Suggestions

In addition to fixing the limitations listed above, below is a non-exhaustive list of how the grammar can be extended:

1. Model the phenomenon where an aspect marker can be inserted to the middle of a verb (e.g. "唱了歌", "睡了觉")
2. Passives (e.g. "他被狗追了。")
3. Relative marker (e.g. "可爱的猫")
4. Accusative marker (e.g. "他把盘子打坏了。")
5. Wh-questions (e.g. "他为什么不吃饭？")
6. Cleft sentences (e.g "她是昨天买的。")
7. Correlative conjunctions (e.g. "因为 ... 所以 ...")
8. Exclamative particles (e.g. "呀，你来了！")
9. Reduplication (e.g. "舒舒服服")
10. Idioms (e.g. "津津有味", "豁然开朗")

Tools

- LinGO Grammar Matrix
- ACE
- LUI
- PyDelphin
- art
- FFTB

Useful commands

Compilation

```
ace -G cmn.dat -g ace/config.tdl
```

Parsing

```
ace -g cmn.dat -l
```

Testing

Create test skeleton

```
mkdir tsdb/skeletons/testsuite
cp tsdb/skeletons/Relations tsdb/skeletons/testsuite/relations
./make_item data/testsuite tsdb/skeletons/testsuite/item
```

Create test profile

```
delphin mkprof -s tsdb/skeletons/testsuite/ trees/testsuite/
```

Populate test profile

```
delphin process -g cmn.dat trees/testsuite/
```

Fetch true positives

```
delphin select 'i-id i-input where i-wf = 1 and readings > 0' trees/testsuite/
```

Fetch true negatives

```
delphin select 'i-id i-input where i-wf = 0 and readings = 0' trees/testsuite/
```

Fetch false positives

```
delphin select 'i-id i-input where i-wf = 0 and readings > 0' trees/testsuite/
```

Fetch false negatives

```
delphin select 'i-id i-input where i-wf = 1 and readings = 0' trees/testsuite/
```

Treebanking

Create gold profile

```
delphin mkprof -s tsdb/skeletons/testsuite/ tsdb/gold
```

Populate gold profile

```
art -f -a 'ace --disable-generalization -g cmn.dat -O' tsdb/gold
```

Launch interactive treebanking interface

```
fftb -g cmn.dat --browser --webdir ~/bin/acetools-x86-0.9.31/assets/ tsdb/gold
```

Generation

Generate in same language

```
echo "<sentence>" | ace -g cmn.dat -Tfq | ace -g cmn.dat -e
```

Translate to another language

```
echo "<sentence>" | ace -g cmn.dat -Tf1 | python <filter_rules>.py | ace -g
<other_language>.dat -e --disable-subsumption-test
```