SGML Syntax Coding for the Map Task corpus

David McKelvie

April 25, 2001

1 Introduction

This chapter describes the SGML markup of the parsed Map Task corpus. The Document Type Description described here annotates only phrasal constituents, as used when `output_parse(phrasal)` is specified.

2 Top level elements

The document element is `parsedchannel`, which contains a `TEXT` element, followed by an optional `SUMMARY` element. The `TEXT` element contains the parsed conversation (for giver or follower) and the `SUMMARY` element gives a summary of how many times particular grammar rules were used, and how many paragraphs of the corpus were parsed into phrases.

```
<!ELEMENT parsedchannel (TEXT,SUMMARY?)>
<!ATTLIST parsedchannel
        role    (Follower|Giver) #IMPLIED
        date    CDATA            #REQUIRED -- Date/time when this file made --
        id      ID               #REQUIRED >
```

TEXT is made up of a sequence (normally alternating) of PARA and GAP. TEXT is made up of a sequence (normally alternating) of PARA and GAP elements. PARAs consist of parsed speech, GAPs are sequences of pauses and noises.

```
<!ELEMENT TEXT ((%markupStuff)*,(PARA|GAP)*) +(MARKER)>

<!ENTITY % markupStuff "UCODE|SCODE|TIMESTAMP|TURNANCHOR|FOREIGN|
                        CITED-WORD|UNCLEAR|VOCAL|EDITORIAL" >
```

3 The structure of PARA

PARA is used as the initial segmentation of the speech. They are the segments used by the POS tagger (ie split before pauses). They have a faint resemblence to paragraphs or sentences.

```
<!ENTITY % paraContent "(%markupStuff;|%wordStuff;|%phraseStuff;)*">
<!ELEMENT PARA  %paraContent >
<!ATTLIST PARA
     TYPE (OK|NOTOK|UTT|TRIVIAL)    OK
     N    NUMBER                    #IMPLIED
     ID   ID                        #IMPLIED >
<!ENTITY % wordStuff   "TW|SIL|NOI|PUN" >
<!ENTITY % phraseStuff "UTT|S|NP|PP|AP|VP|ADVP|STARTER|CLOSER" >
```

N is a number (starting from 1) which counts the number of the paragraph in the file. TYPE describes how the text has been parsed:

TRIVIAL : Paragraph consists only of noises, pauses, discourse markers, filled pauses, interjections and punctuation.

UTT : Paragraph has been parsed into a single UTT element.

OK : Paragraph has been parsed into a sequence of phrasal level constituents.

NOTOK : Paragraph contains non-phrasal constituents not contained in any higher constituent.

We may have an initial timestamp outside a gap, otherwise timestamps are inside para or gap.

GAP consists of a sequence (possibly empty) of NOI followed by either sentence final punctuation or more likely a sentence tagged silence.

```
<!ENTITY % gapContent     "(SIL|NOI|%markupStuff;)*" >
<!ELEMENT GAP  %gapContent >
```

4 Phrase level elements

4.1 Phrase structure

The attributes defined for these phrase level elements are defined below.

Utterances

```
<!ENTITY % uttContent  "%paraContent" >
<!ELEMENT UTT   %uttContent >
<!ATTLIST UTT %start %dur %rule>
```

Sentences

```
<!ENTITY % SContent    "%paraContent" >
<!ELEMENT S     %SContent >
<!ATTLIST S %vform %dum %abort %conj %inv %start %dur %rule>
```

Phrasal Noun Phrases:

```
<!ENTITY % NPContent   "%paraContent" >
<!ELEMENT NP %NPContent >
<!ENTITY % Nattributes "%num %pers %case %abort %dum %type %sem %wh %start %dur %rule" >
<!ATTLIST NP %Nattributes >
```

Prepositional Phrases:

```
<!ENTITY % PPContent   "%paraContent" >
<!ELEMENT PP %PPContent >
<!ATTLIST PP %prep %conj %abort %wh %start %dur %rule>
```

Verb Phrases:

```
<!ENTITY % VPContent   "%paraContent" >
<!ELEMENT VP %VPContent >
<!ATTLIST VP %vform %num %pers %clitic %abort %aux %mod %tns %start %dur %rule>
```

Adverb Phases:

```
<!ENTITY % ADVPContent "%paraContent" >
<!ELEMENT ADVP %ADVPContent >
<!ATTLIST ADVP %abort %deg %wh %start %dur %rule>
```

Adjective Phrases:

```
<!ENTITY % APContent   "%paraContent" >
<!ELEMENT AP %APContent >
<!ATTLIST AP %abort %deg %wh %start %dur %rule>
```

Starters:

```
<!ENTITY % StarterContent "%paraContent" >
<!ELEMENT STARTER %StarterContent >
<!ATTLIST STARTER %abort %start %dur %rule>
```

Closers:

```
<!ENTITY % CloserContent  "%paraContent" >
<!ELEMENT CLOSER %CloserContent >
<!ATTLIST CLOSER %abort %start %dur %rule>
```

4.2 Definitions of phrasal attributes

```
<!ENTITY % case   "CASE   (subj|obj|undef_case)          obj">
```

Case of nouns: SUBJ is nominative case (subject), OBJ is objective case (object), UNDEF_CASE is when case is uncertain or not determined by parser.

```
<!ENTITY % deg    "DEG    (pos|comp|sup|undef_deg)       pos">
```

Degree of adjectives or adverbs: POS is positive (i.e. base form of adjectives), COMP is comparative form (e.g. “bigger”), SUP is the superlative form (e.g. “biggest”), UNDEF_DEG is when parser has not determined the degree.

```
<!ENTITY % dur    "DUR    NUTOKEN                        0">
```

The duration of the word or phrase in seconds.

```
<!ENTITY % num    "NUM    (sing|plur|undef_num)          sing">
```

Number of nouns: SINGular, PLURal or undetermined.

```
<!ENTITY % pers   "PERS   (1|2|3|undef_pers)             3">
```

The person of nouns:

```
<!ENTITY % vform  "VFORM  (fin|en|ing|bse|to|imp|
                           undef_vform)                  fin">
```

The form of a verb:

FIN : finite form, e.g. “goes”.

EN : past participle form e.g. “gone”.

ING : present participle form e.g. “going”.

BSE : base infinitive form e.g. “go”.

TO : "to" infinitive form e.g. "to go".

IMP : imperative form e.g. "go!".

UNDEF_VFORM : used when parser is unsure which form to assign.

```
<!ENTITY % wh     "WH     (wh|not_wh|undef_wh)           not_wh">
```

Whether a noun starts with a "wh" word, e.g. "which book".

```
<!ENTITY % abort  "ABORT  (abort|not_abort|undef_abort)  not_abort">
```

Whether a phrase is aborted(unfinished) or not.

```
<!ENTITY % aux    "AUX    (aux|not_aux|undef_aux)        not_aux">
```

Used for verbs to specify whether it is an auxiliary verb or not.

```
<!ENTITY % clitic "CLITIC (clitic|not_clitic)            not_clitic">
```

Whether a verb is a clitic or not e.g. "+'s" versus "is".

```
<!ENTITY % dum    "DUM    (dum|not_dum|undef_dum)        not_dum">
```

Whether a noun is a dummy noun such as "there" or not.

```
<!ENTITY % conj   "CONJ   CDATA                          none">
```

Used if a S or PP starts with a conjunction, if so the lexical value of the conjunction is stored as the value of this attribute.

```
<!ENTITY % inv    "INV    (inv|not_inv|undef_inv)        not_inv">
```

Is a S inverted or not? e.g. "is it time for tea?" is inverted.

```
<!ENTITY % type   "TYPE   (norm|pro|prop)                norm">
```

Type of noun phrase, PRO is a pronoun, PROP are proper nouns, NORM all the rest.

```
<!ENTITY % rule   "RULE   CDATA                          #IMPLIED">
```

The identifier of the rule which constructed this constituent.

```
<!ENTITY % sem    "SEM    (measure|any|undef_sem)        any">
```

Noun phrases are marked if they are measure phrases or not, e.g. "three inches" is a measure phrase, "two trees" is not.

```
<!ENTITY % start  "START  CDATA                          'UNDEF_START'" --undefined -->
```

Start time of element, in seconds from start of file.

```
<!ENTITY % prep   "PREP   CDATA                          #REQUIRED">
```

The head proposition in a PP.

```
<!ENTITY % mod    "MOD    (mod|not_mod|undef_mod)        not_mod">
```

Whether a VP has a modal verb as head or not.

```
<!ENTITY % tns    "TNS    (pres|past|undef_tns)          pres">
```

The tense of finite verb phrases.

5 The structure of the summary

```
<!ELEMENT SUMMARY (PARAS,TRIVIAL,NONTRIVIAL,UTTS,OK,NOTOK,RULE*)>
```

PARAS is the number of <PARA> elements in the file.

TRIVIAL is the number of these which are trivial.

NONTRIVIAL are the other non-trivial ones.

UTTS is the number which have been parsed as <UTT>.

OK is the number which have parsed as a sequence of reasonable pieces.

NOTOK are the rest which have not been well parsed.

RULE counts how many times each rule was used

In the last three elements i.e. UTTS, OK, and NOTOK; the PERC attribute is the percentage of all <PARA>s and NTPERC is percentage of the non-trivial ones.

```
<!ELEMENT (PARAS, TRIVIAL, NONTRIVIAL, UTTS, OK, NOTOK, RULE) - o EMPTY>
<!ATTLIST PARAS
    N       NUMBER   0 >
<!ATTLIST (TRIVIAL, NONTRIVIAL)
    N       NUMBER   0
    PERC    NUTOKEN  0 >
<!ATTLIST (UTTS, OK, NOTOK)
    N       NUMBER   0
    PERC    NUTOKEN  0
    NTPERC  NUTOKEN  0 >
<!ATTLIST RULE
    N       NUMBER   0
    R       CDATA    #REQUIRED >
```

For example, a summary element element might look like:

```
<SUMMARY>
<PARAS      N = 48>
<TRIVIAL    N = 9 PERC=18.75>
<NONTRIVIAL N = 39 PERC=81.25>
<UTTS       N = 29 PERC=60.42 NTPERC=74.36>
<OK         N = 5 PERC=10.42 NTPERC=12.82>
<NOTOK      N = 5 PERC=10.42 NTPERC=12.82>
<RULE R="main(s11)" N=1>
<RULE R="disf(rp(a))" N=2>
<RULE R="main(advp06)" N=1>
<RULE R="disf(main(s06))" N=1>
<RULE R="main(abort(np02))" N=1>
<RULE R="disf(utt(1))" N=1>
<RULE R="disf(main(hack1))" N=1>
<RULE R="main(vpintrans)" N=1>
<RULE R="main(ap16)" N=2>
<RULE R="main(np6catch)" N=2>
...
</SUMMARY>
```

6 Etc

Other miscellaneous SGML markup.

```
<!ELEMENT       foreign - -     (%paraContent)>
<!ATTLIST       foreign
        lang    (FR | DE | LA | ES)     #REQUIRED>

<!ELEMENT       unclear - -     (%paraContent)>
<!ELEMENT       cited-word      - -     (%paraContent)>

<!ENTITY amp "&">
<!ENTITY noise "&amp;noise">

<!ELEMENT turnanchor            EMPTY>
<!ATTLIST turnanchor    synch   NAME    #REQUIRED > <!-- really an idref-->
```