The mapping between the parsing annotation schemes of the Lancaster Parsed Corpus and the Susanne Corpus

Hong Liang Qiao
University of Queensland

1. Introduction

Many corpora have been tagged and parsed with different tagging and parsing schemes and are not mutually usable because of such differences. The main aim of a mapping is that parsed corpora with different annotations can be changed to a version which is annotated with the required scheme so as to extend the research resources. In other words, the mapping between annotation schemes will enable the reusability of the resources of the existing tagged and parsed corpora.

To do the mapping manually or semi-automatically from one scheme to another corpus of a certain size is time-consuming and complex. Generally we map from a more sophisticated parsing scheme to a less sophisticated one, simply because it is much more difficult or in most cases impossible to achieve the reverse.

This paper discusses an attempt to map between the parsing annotations of two parsed corpora – the Susanne Corpus (see Sampson 1992) and the Lancaster Parsed Corpus (see Garside 1993). The direction is from Susanne to the LPC. Eighty-eight items of the annotation are involved in the research and thirty-three are focused on as typical examples for detailed investigation and explanation, out of which five situations are found and explained to confirm the feasibility of such mapping technically. Only typical examples are used in the explanations of the mapping. Not all of them are dealt with in detail in the paper. Some results are shown in Appendix C. The approach described in this paper will serve as a model for the further mapping between the parsing annotation schemes of parsed corpora.

2. The two annotation schemes

Both the Susanne Corpus and the Lancaster Parsed Corpus are tagged and structurally parsed. The Susanne Corpus is also provided with function tags. The mapping here focuses on the parsing annotation schemes. The tagging and the function annotations will not be discussed in this paper.

2.1. The Susanne Parsing Scheme

The Susanne Parsing Scheme has formtags on three levels, namely the root level, clause level and phrase level. The formtags are normally represented by an upper case letter, which stands for a broad classification, while some take a lower case letter to indicate a subcategory.

At root level, there are formtags such as “Q” for quotation, “I” for interpolation and “Iq” for tag question. At clause level, “S” stands for main clause, while “Ss” stands for a reporting clause embedded within the quotation. At phrase level, “N” is a noun phrase as a category and a lower case letter can be attached to it to represent a subcategory of the noun phrase. For example, “Ns” is a singular noun phrase and “Np” is a plural noun phrase.

Both the upper case letter and the lower case letter are usually the initial letter of certain grammatical terms and they can be quite easily recognised and remembered. For example, “P” stands for prepositional phrase and “Po” and “Pb” respectively stand for prepositional phrases started with *of* and *by*. However, not all formtags are as meaningful as these. For details of the Susanne Parsing Scheme, see Appendix A.

2.2. The LPC Parsing Scheme

The LPC Parsing Scheme divides its parsing tags into five types, i.e. “sentence tags”, “finite clause tags”, “non-finite and verbless clause tags”, “constituent tags for major phrase types” and “constituent tags for minor phrase types”. Like the Susanne Parsing Scheme, the LPC parsing tags are also made up of either an upper case letter or an upper case plus a lower case letter. Usually the upper case letter represents the general category of a grammatical classification, while the lower case letter stands for a detailed further category within the general category.

As in the Susanne Parsing Scheme, the LPC parsing tags are also meaningful and quite easy to recognise. Details are given in Appendix B. The mapping between the two schemes will be discussed in Section 4.

3. The methods

Basically, there are two methods for the mapping:

a) Look through the lists of the parsing schemes, i.e. Appendix A and B, compare the names of the phrase and clause tags, their definitions and examples to see if they belong to one of the following situations:

- same
- similar
- different

b) If it is not clear from the above comparison, find actual words or phrases that belong to a certain phrase or clause of one corpus in the other corpus. If the words or phrases mean the same and function in the same way, the two tags are then matched. This method can also provide evidence for the first method above.

4. Mapping between the two parsing schemes

The mapping between the parsing annotation schemes of the two corpora is unidirectional from Susanne to the LPC – that is, all Susanne annotations are meant to be changed to those of LPC. In this paper, only the five main situations summed up from the investigation are presented with examples. A table showing the results of the mapping is attached in Appendix C.

The five main situations are discussed in detail below, and application methods for the computer in executing the mapping are presented as well.

4.1. The formtags with the same name

In the first situation there are two types. The first type represents formtags which have the same name and are equal. In the second type the formtags have same name, but one is included in the other, i.e. Susanne's formtags are included in those of LPC. The solutions proposed for each type may be used for a computer automatic mapping or a manual mapping if it is complicated enough.

4.1.1. Equal (Type 1:1)

In the first type both Susanne and LPC have the same formtag with the same grammatical definition. For example, "Ti" (*to*-infinitive clause) and "W" (non-finite or verbless clause introduced by *with*), etc. fall in this category and they are exactly the same not only in name, but also in meaning and grammatical analysis. This group represents more than 1/5 of the total formtags to be mapped.

The solution to Type 1:1 is that no change of the Susanne formtags is necessary.

4.1.2. Subordinate (Type 1:2)

In the second type we have the same formtag, but the Susanne formtag is included in the LPC formtag. For instance, "S" (main clause) in LPC not only covers "S" (main clause excluding subjunctive clause, interrogative clause, etc.) in Susanne, but also "S!" (exclamatory clause), "S?" (interrogative clause), "S%" (subjunctive clause), and "S*" (imperative clause). In other words, the two "S"s are not equal to each other.

The solution to Type 1:2 is to change the Susanne formtags to those of LPC.

4.2. Different names

The second situation is that the two corpora have different formtags for the same or similar grammatical items. This is a fairly complicated situation, in which there are four types of mappings and several ways to solve the problem accordingly.

4.2.1. Equal (Type 2:1)

The first type requires Susanne's formtags to change to those of LPC on a one-to-one basis, since the two schemes have different formtags for the same grammatical item, though superficially they do not resemble each other. The method to process this mapping is to replace the Susanne tag with its counterpart in LPC. Here is an example:

"Sq" in LPC means "a piece of quotation normally an independent piece of language which occurs in fictional dialogue enclosed in quote marks", for example:

a) "Nothing will change my mind," said Pat.
   [S "[Sq]", [V] [N]S]

b) Pat said, "Nothing will change my mind".

[S [N] [V], “[Sq]” S]

“Q” in the Susanne Corpus is a root-level formtag indicating quotation, although “Q” is a node above S.

A01:0490b	–	YIL	<ldquo>	–	[O[S.
A01:0490c	**–**	**DD2i**	**+These**	**these**	**[Q:o[S[Np:s.**
A01:0490d	**–**	**NN2**	**actions**	**action**	**.Np:s]**
A01:0490e	**–**	**VMd**	**should**	**shall**	**[Vdc.**
A01:0490f	**–**	**VV0v**	**serve**	**serve**	**.Vdc]**
A01:0490g	**–**	**TO**	**to**	**to**	**[Ti:z[Vi.**
A01:0490h	**–**	**VV0t**	**protect**	**protect**	**.Vi]**
A01:0490i	**–**	**II**	**in**	**in**	**[P:h.**
A01:0490j	**–**	**NN1n**	**fact**	**fact**	**.**
A01:0490k	**–**	**CC**	**and**	**and**	**[P+.**
A01:0490m	**–**	**II**	**in**	**in**	**.**
A01:0490n	**–**	**NN1n**	**effect**	**effect**	**.P+]P:h]**
A01:0500a	**–**	**AT**	**the**	**the**	**[Np:o[G[Ns.**
A01:0500b	**–**	**NNJ1n**	**court**	**court**	**.Ns]**
A01:0500c	**–**	**GG**	**+<apos>s**	**–**	**.G]**
A01:0500d	**–**	**NN2**	**wards**	**ward**	**.Np:o]**
A01:0500e	**–**	**II**	**from**	**from**	**[P:r.**
A01:0500f	**–**	**JJ**	**undue**	**undue**	**[Np.**
A01:0500g	**–**	**NN2**	**costs**	**cost**	**.Np]P:r]**
A01:0500h	**–**	**CC**	**and**	**and**	**[Ti+.**
A01:0500i	**–**	**APPGh1**	**its**	**its**	**[Np:o.**
A01:0500j	**–**	**VVNt**	**appointed**	**appoint**	**[Tn[Vn[VVNt&.**
A01:0500k	**–**	**CC**	**and**	**and**	**[VVNt+.**
A01:0500m	**–**	**VVNt**	**elected**	**elect**	**.VVNt+]VVNt&]Vn]Tn]**
A01:0510a	**–**	**NN2**	**servants**	**servant**	**.Np:o]**
A01:0510b	**–**	**II**	**from**	**from**	**[P:r.**
A01:0510c	**–**	**JJ**	**unmeritorious**	**unmeritorious**	**[Np.**
A01:0510d	**–**	**NN2**	**criticisms**	**criticism**	**.Np]P:r]Ti+]Ti:z]S]Q**:o]
A01:0510e	–	YIR	+<rdquo>	–	.
A01:0510f	–	YC	+,	–	.
A01:0510g	–	AT	the	the	[Ns:s.
A01:0510h	–	NN1c	jury	jury	.Ns:s]
A01:0510i	–	VVDv	said	say	[Vd.Vd]S]
A01:0510j	–	YF	+.	–	.O]

Now we may come to the conclusion that when in Susanne there is a pattern x as follows:

x) [S“[Q]”, [N] [V] S]

it can be transcribed as pattern a):

a) [S “[Sq]”, [V] [N] S]

If the quotation is after the main verb, pattern b) is allowable:

b) [S [N] [V], "[Sq]" S]

The solution to Type 2:1 is to change Susanne's "Q" to the LPC's "Sq". N.B. "Q" may occur with a colon-initiated suffix. "Q" should also be placed after "S" in the LPC. This needs either manual mapping or a small program to handle it.

4.2.2. Subordinate (Type 2:2)

There is a second type of mapping within this situation which is the most common one – a few or sometimes many Susanne formtags match one LPC formtag. One thing observable in such a mapping is that many of the Susanne formtags have to take a broader-sense LPC formtag, due to the fact that the Susanne scheme is much more detailed in design than the LPC's. It is no longer a one-to-one mapping in this case. Instead, it is a several-to-one mapping. In this way, the computer should just replace the Susanne formtags with the LPC one, though they do not use the same name.

"Ff" in the Susanne scheme means a "fused" relative clause. In their manual, Garside *et al* (1993:12) said, "a 'fused' or 'nominal relative clause' as in 'I will do [what you want]' is treated as 'Fn'". The definition of an "Fn" is "a finite nominal clause, such as a finite subordinate clause which functions in the position of a noun phrase. Examples of 'Fn' are *that*-clauses and *wh*-clauses (including indirect statements and indirect questions, also including 'zero *that*-clauses', where the *that* is omitted at the beginning of the clause), e.g.: 'I know [Fn that you saw them Fn]'." "Ff" can be covered by "Fn" in LPC.

The solution to Type 2:2 is to replace Susanne's "Ff" with LPC's "Fn".

4.2.3. Reverse subordinate (Type 2:3)

The third type is that a Susanne formtag matches two or more LPC counterparts, one of which is the same name and the other(s) different. Only one case of this type was found in the mapping.

"Ms" in the Susanne scheme means a numeric phrase headed by *one*. It very often takes the form of a *one of* ... phrase. However, in the LPC it is a noun phrase. Example of "Ms" – *one of* in Susanne:

```
A05:0170p   –   MC1     One      one      [S[S:o[Ms:s.
A05:0170q   –   IO      of       of       [Po.
A05:0170r   –   DD2i    these    these    [Np.
A05:0180a   –   NN2     men      man      .Np]Po]Ms:s]
```

Example of “N” – *one of* in LPC:

F06 378
[S&[N many_AP architects_NNS [Po of_INO [N our_PP$ acquaintance_NN N]Po]N][V would_MD dissent_VB V][P from_IN [N this_DT last_AP view_NN N]P] ,_, [S+ but_CC [N the_ATI fact_NN N][V remains_VBZ V][Fn that_CS [N fleas_NNS N][V can_MD still_RB be_BE V]**[N one_CD1 [Po of_INO [N the_ATI main_JJB hazards_NNS [Po of_INO [Tg[Vg lying_VBG Vg][P in_IN [N bed_NN N]P]Tg]Po]N]Po]N]**Fn]S+] ._. S&]

The solution to Type 2:3 is that we should pay attention to whether the “Ms” is followed by an *of* or other “IN”s (prepositions). If yes, then it belongs to “N”; if not, it is an “M”, which means numeric phrase – that is, not all “Ms” can automatically be converted to “N”. In the above example, Susanne’s “Ms” equals LPC’s “N”.

4.2.4. Multiple (Type 2:4)

In the fourth type a formtag in Susanne is combined with different function tags, which creates several variants. The variants may match several different formtags in LPC. If we view Susanne’s formtags, regardless of function tags, as a *group* of formtags or one general formtag on the formtag level (not wordtag), it is then a one (Susanne) to several (LPC) mapping; otherwise, the mapping can be in a several to several form. The computer should then focus on the function tags when dealing with the mapping of a formtag of this kind, and not just process it as a general formtag, while all others mentioned in this section need no attention to the function tags at all.

“Dp”, a plural determiner phrase of the Susanne Parsing Scheme, is a similar case to “Ds”. “Dp”s are found with variants, such as “Dp:s”, “Dp:S”, “Dp:o” and “Dp:e”. Adding the original form “Dp”, there are five types. “Dp” does not have the two other functions “Ds” has, namely “Ds:i” and “Ds:h”. Simple “Dp” without any function tags attached to it very often occurs in “N” and “P” with another “P” embedded in it, rather than acting as subjects and objects in a sentence, because otherwise they are “Dp:s”s.

#1 “Dp” in “N”:

G01:1650m	–	CC	and	and	[S+.
G01:1650n	**–**	**XX**	**not**	**not**	**[Nnp:s[Dp.**
G01:1660a	**–**	**DA2**	**many**	**many**	**.Dp]**
G01:1660b	**–**	**NP2s**	**Bourbons**	**Bourbon**	**.Nnp:s]**
G01:1660c	–	VV0t	deny	deny	[V.V]
G01:1660d	–	PPHO2	them	they	[Nop:o.Nop:o]S+]S]

Compared to LPC:

K02 47
*'_*' [S[E there_EX E][V are_BER V]**[N[D far_RB too_QL many_AP D] double-barrelled_JJ names_NNS N]**[R out_RP R][R here_RN R][Fa as_CS [N it_PP3 N][V is_BEZ V]Fa] ._. S] **'_**'

So Susanne's "Dp" is actually LPC's "D". N.B. If there is only one determiner in "N", it is not counted as a "Dp" or "D" of any kind, e.g.:

A01 75
[S[N it_PP3 N][V had_HVD offended_VBN V][N **many_AP** people_NNS [P far_RB beyond_IN [N the_ATI ranks_NNS [Po of_INO [N labour_NN supporters_NNS N]Po]N]P]N] ._. S]

#2 "Dp" in a "P" with another "P" embedded in "Dp" after the determiner head:

Susanne's "Dp" in "P":

J22:0100d	–	II	in	in	[S-[P:p.
J22:0100e	**–**	**DDi**	**some**	**some**	**[Dp.**
J22:0100f	**–**	**IO**	**of**	**of**	**[Po.**
J22:0100g	**–**	**AT**	**the**	**the**	**[Np.**
J22:0100h	**–**	**JJ**	**new**	**new**	**.**
J22:0100i	**–**	**NN2**	**nations**	**nation**	**.Np]Po]Dp]**P:p]

LPC's "N" in "P":

A07 477
*'_*' [S[N it_PP3 N][V was_BEDZ V][P before_IN [N the_ATI 1957-58_CD-CD tour_NN [Po of_INO [N South_NP Africa_NP ,_, [Fr[Rq when_WRB Rq][N Bagenal_NP N][V said_VBD V][R half-jokingly_RB R]**[P before_IN [N some_DTI [Po of_INO [N the_ATI team_NN N]Po]N]P]**Fr]N]Po]N]P] :_: S]

Susanne's "Dp" has to be changed to LPC's "N" in this case.

#3 "Dp:s" – determiner phrase as subject:

A08:1820f	**–**	**DDi**	**some**	**some**	**[Dp:s.**
A08:1820g	**–**	**IO**	**of**	**of**	**[Po.**
A08:1820h	**–**	**APPGm**	**his**	**his**	**[Np.**
A08:1820i	**–**	**NN2**	**followers**	**follower**	**.Np]Po]Dp:s]**

R01 85
[S[V let_VB V][N us_PP1OS N][Tb[V examine_VB V][P in_IN [N detail_NN N]P]**[N some_DTI [Po of_INO [N the_ATI Jones_NP policies_NNS [P for_IN [N Britain_NP N]P]N]Po]N]**Tb] :_: S]

“Dp:s” should be changed to LPC’s “N”.

#4 “Dp:S” – determiner as “surface (and not logical) subject”:

G04:1360h	–	CST	that	that	[Fc.
G04:1360i	**–**	**DBa**	**all**	**all**	**[Dp:S179.**
G04:1360j	**–**	**IO**	**of**	**of**	**[Po.**
G04:1360k	**–**	**AT**	**the**	**the**	**[Np.**
G04:1360m	**–**	**NN2**	**objects**	**object**	**.Np]Po]Dp:S179]**
G04:1370a	–	VVDi	seemed	seem	[Vd.Vd]

“Dp:S” is nonetheless the same as “Dp:s” in (*30) #3, so we replace it with LPC’s “N”.

#5 “Dp:o” – determiner phrase as logical direct object:

01:0650m	–	CC	and	and	[S+.
G01:0660a	–	VVNi	come	come	[Vn.Vn]
G01:0660b	–	RR	close	close	[R:q.
G01:0660c	–	IIt	to	to	[P.
G01:0660d	–	VVGt	wrecking	wreck	[Tg[Vg.Vg]
G01:0660e	**–**	**DA2q**	**several**	**several**	**[Dp:o.**
G01:0660f	**–**	**DAR**	**more**	**more**	**.Dp:o**Tg]P]R:q]S+]S+]

K10 1114
’_’ [S&[Na I_PP1A Na][V did_DOD n’t_XNOT bring_VB V][N you_PP2 N][N flowers_NNS N] ,_, [N Magda_NP N] ,_, [Fa because_CS [Na I_PP1A Na][V know_VB V][Fn[N you_PP2 N][R always_RB R][V have_HV V]**[N so_QL many_AP N]**Fn] *-_*- [S-[Na we_PP1AS all_ABN Na][V do_DO V]S-]Fa] ._. S&]

Susanne’s “Dp:o” is treated as LPC’s “N”.

#6 “Dp:e” – determiner phrase as predicate complement of the subject:

N05:1030d	–	CC	And	and	[S+.
N05:1030e	–	MC1	one	one	[Ms:s.Ms:s]

N05:1030f	–	VHD	had	have	[Vdfb.
N05:1030g	–	VBN	been	be	.Vdfb]
N05:1030h	**–**	**RGf**	**too**	**too**	**[Dp:e.**
N05:1030i	**–**	**DA2**	**many**	**many**	**.Dp:e]S+]**

B03 164
[S&[N a_AT thousand_CD delegates_NNS N][V are_BER V]**[N[D too_QL many_AP D]N]**[P for_IN [N corporate_JJ thinking_NN N]P] ,_, [S+ but_CC [N corporate_JJ thinking_NN N][E there_EX E][V must_MD be_BE V][Fa if_CS [N all_ABN member_NN churches_NNS N][V are_BER V][Ti[Vi to_TO have_HV Vi][N an_AT effective_JJ voice_NN [P in_IN [Tg[Vg deciding_VBG Vg][N future_JJB lines_NNS [Po of_INO [N cooperation_NN N]Po]N]Tg]P]N]Ti]Fa]S+] ._. S&]

“Dp:e” matches “N” in LPC.

The solution to Type 2:4 is to change from Susanne’s formtags to those of LPC as follows:

“Dp”	= “D” in “N”
“Dp”	= “N” in “P”
“Dp:s”	= “N”
“Dp:S”	= “N”
“Dp:o”	= “N”
“Dp:e”	= “N”

At the same time, attention should be paid to some conditions in the mapping, such as in “Dp” = “D” in “N” and “Dp” = “N” in “P”.

Therefore, in this situation there are four parallel mapping forms in situation 2, i.e. one to one, several to one, one to several and several to several. There is at the same time a matter of mapping directions as illustrated in Figure 1 (directions not listed are not applicable),

	Situation 2	LPC	Directions	Susanne
	type 1	one	<-- -->	one
	type 2	one	-->	several
	type 3	several	<--	one
	type 4	several	<--	one group
or:	type 4	several	<-- -->	several (in one group)

Fig. 1. The mapping types in Situation 2

4.3. No counterpart in the LPC or zero replacement (Type 3:1)

The third situation is when there is a certain formtag in Susanne, and no such formtag in LPC. The method to deal with this is to make a zero replacement or to delete the formtag in the corpus. However, this is a minor type in the mapping.

“Jx” – “measured absolute J”, consists of two or more hyphenated words as an adjective. See the Susanne example below:

A11:1400j –AT1 a a [Ns@.

A11:1400k	**–**	**MCn**	**3**	**–**	**[Jx[N.**
A11:1400m	**–**	**YH**	**+hyphen**	**–**	**.**
A11:1400n	**–**	**NNT1c**	**+year**	**year**	**.N]**
A11:1400p	**–**	**YH**	**+hyphen**	**–**	**.**
A11:1400q	**–**	**JJ**	**+old**	**old**	**.Jx]**
A11:1400r	–	NN1c	filly	filly	.Ns@]

But in LPC, there is no “Jx”. Instead, the collective hyphenated multi-word adjective is assigned a wordtag “JJB”:

A02 117
[S[R second_RB [P in_IN [N command_NN N]P]R][V is_BEZ V][N \0Mr_NPT Eric_NP Roll_NP ,_, [N **53-year-old_JJB** deputy_NN Secretary_NPT [P at_IN [N the_ATI ministry_NN [Po of_INO [NN/NN/NNS& agriculture_NN ,_, [NN- food_NN ,_, NN-] [NNS+ and_CC fisheries_NNS NNS+]NN/NN/NNS&]Po]N]P] N] N] ._. S]

Since “Jx” is the major formtag in Type 3:1, the solution to Type 3:1 is first: the formtag “Jx”, “N” and the parts-of-speech tags of *3-year-old* should be deleted. Then, instead of putting a new formtag in the position of “Jx”, a wordtag is assigned to the three hyphenated words as a whole. “Jx” is quite a complicated case in the mapping, and a manual mapping is preferred in this particular situation. However, “Jx” is not always a hyphenated sequence in the Susanne Corpus.

4.4. No counterpart in Susanne (1) (Type 4:1)

The fourth situation mainly concerns the difference between the annotations of *not* in Susanne and LPC. There is no “X” in Susanne. The computer program is required to recognise and then pick out the *not* which is not the adverb in a verb phrase of any kind and assign an “X” formtag to it. It is more than a simple string for string matching.

“X” is the negative word *not* when acting as an independent element of clause structure; e.g. in “He told us [Fn what not to do Fn]”, *not* follows the object of the subordinate clause introduced by *what* and precedes the verb phrase *to do*. Thus, the clause *what not to do* has the three constituents “Nq”, “X”, and “Vi”. Generally, *not* is part of the verb phrase (see under “V” above) and therefore does not require an “X” as a special formtag in LPC, but definitely in other places, where “X” is needed, e.g:

LPC's *not* outside the verb phrase:

D01 4
[S[N some_DTI critics_NNS N] ,_, [Si**[X not_XNOT X]**[N many_AP N]Si] ,_, [V argue_VB V][Fn that_CS [N the_ATI gospel_NN N][V is_BEZ V][N the_ATI product_NN [Po of_INO [N& one_CD1 mind_NN [N+ and_CC one_CD1 hand_NN N+] N&]Po]N]Fn] ._. S]

LPC's *not* in the verb phrase:

A01 24
[S[N \0Mr_NPT Macleod_NP N]**[V was_BEDZ not_XNOT V]**[P at_IN [N the_ATI week-end_NN meeting_NN N]P] ._. S]

Not as not part of a verb phrase in Susanne:

N01:0400m	–	JJ	young	young	[J:e.
N01:0400n	–	YC	+,	–	.
N01:0400p	**–**	**XX**	**not**	**not**	**[D-.**
N01:0400q	–	DAR	more	more	.
N01:0400r	–	CSN	than	than	[P.
N01:0400s	–	MC	nineteen	nineteen	[M.
N01:0410a	–	CCr	or	or	[M+.
N01:0410b	–	MC	twenty	twenty	.M+]M]P]D-]J:e]Fn:o]S+]S]

Not as part of the verb phrase:

A01:0550d	–	VDD	did	do	[Vde.
A01:0550e	**–**	**XX**	**not**	**not**	**.**
A01:0550f	–	VV0t	elaborate	elaborate	.Vde]

The solution to Type 4:1 is that "not (XX)" in Susanne is NOT always embedded in a verb phrase of any kind. The computer should be able to find out that, from the point *not*, there should be no "V" formtags closed further down before the opening of another formtag. Only in that case is Susanne's *not (XX)* an "X" in LPC. Therefore "X" should be on both sides of *not* in the Susanne Corpus. Otherwise it will be the *not* in a verb phrase and then nothing needs to be done. It is therefore not a simple match. It involves a small program to execute the recognition and assignment of the formtag "X", or a manual mapping is necessary.

4.5. No counterpart in Susanne (2) (Type 5:1)

The fifth situation is just the reverse of the second situation and similar to the fourth, in which there is no Susanne formtag matchable to LPC's formtag. Thus, to solve this problem, the lower platform, i.e. the part-of-speech tags, has to be checked. Luckily, the only two items of this situation take the same part-of-speech tags as LPC and what the computer program needs to do is to assign on the left and right side of the item respectively an "[(LPC formtag)" and an "(LPC formtag)]" in the Susanne Corpus.

"E" is the label used for existential *there*, i.e. the unstressed *there* in the *there is/are* construction. E.g. "[E There E] is nothing wrong".

The solution to Type 5:1 is to find "EX", the wordtag for existential *there* and put "[E" on the left and "E]" on the right side of "there_EX". This method is also applicable to "U". Figure 2 sums up the above five situations:

Situations: Types	LPC –	Susanne
1:1	X =	X
1:2	X >	X
2:1	X =	A
2:2	X >	A, B, C, etc.
2:3	X, Y, Z >	A
2:4	X, Y, Z >	A:a, A:b, ... A:n
or: 2:4	X, Y, Z >	A:(a→n)
3:1	X =	"" (zero)
4:1	X =	[X +[A (B) +A] X]
5:1	X =	+[X (A) +X]

Fig. 2. Mapping types of the parsing schemes. "=" means "to be equal to", ">" means "to include", "–" means "to take out", and "+" means "to add".

5. General statistics on the mapping between the two schemes

Altogether, there are nine types of mapping from all the eighty-eight formtags of the Susanne Corpus to thirty-three LPC parsing tags or tags that need special processing. The mapping is classified into nine types in the five situations. Figure 3 shows the general statistics:

Situations & Types	Number of parsing tags	Proportion	Rank
1:1	18	21.6%	2
1:2	12	13.6%	3
2:1	2	2.3%	5
2:2	47	53.4%	1
2:3	1	1.1%	6
2:4	3	3.4%	4
3:1	1	1.1%	6
4:1	1	1.1%	6
5:1	2	2.3%	5
Total 9	88	99.9%	

Fig. 3. Statistics on mapping types

Only three types occur in significant numbers, that is “2:2” has 53.4%, “1:1” 21.6% and “1:2” 13.6%, which make up a proportion of 88.6% out of the total. Therefore, they are the most typical sort of situations and types. They just need simple replacement, which can be executed on a computer fairly easily. Ten formtags in the mapping are found with the “+” sign (see Appendix C), which means that they do not just involve formtag to formtag matching. They need a small program to perform a specific recognition and then assign a relevant formtag, or probably manual conversion is needed with some of them. This type consists of only about 5.7%. The percentage is small and the number of actual occurrences is not expected to be high either.

6. Some mapping techniques resulting from the investigation

One technique is to examine the same two formtags in each corpus and determine, according to the manuals, whether they are identical in definitions and agree with respect to examples. If so, then they are equal to each other. In some cases one may be included in the other, even if they bear the same name.

If formtags have different names, they may match as well. Attention needs to be paid to the fact that one may be equal to or include several others. This may be true of both directions.

If the formtags in one of the schemes cannot be identified or found at all, examples from the corpus should be found. Then the typical ones are picked out and the lexical items are checked (or sometimes together with related wordtags) in the other corpus to discover the formtags which they are assigned. If no formtags are found, then we may have a zero replacement.

Sometimes a Susanne formtag may match different formtags in LPC; in that case, check the function tags it takes and see if they affect the mapping.

Finally, if a formtag cannot simply be replaced, we may have to study the environment of the item. Special programs are needed to process such complicated cases.

7. Conclusions

From the mapping between the parsing schemes of the Lancaster Parsed

Corpus and the Susanne Corpus, we may conclude that the conversion from Susanne parsing tags to the LPC parsing tags seems to be feasible. Much of the work can be done by simple replacement from the Susanne parsing tags to the LPC's, although a comparatively small number of the tags still need extra work. At the same time, there is a need for a grammatical stocktaking (see Sampson 1993). Furthermore, those who are involved in designing corpora for the purpose of natural language processing need to work together to define a standardised parsing scheme that is suitable for most corpora, to prevent problems which may be too late to solve once the corpora have been completed.

Acknowledgments

The author would like to thank Dr. Geoffrey Sampson of the University of Sussex for his advice on the mapping. Thanks also go to Prof. Roland Sussex and Mr. Peter White of the University of Queensland for their valuable comments on the paper.

References

Garside, Roger, Geoffrey Leech, and Geoffrey Sampson (eds.). 1987. *The computational analysis of English: A corpus-based approach*. London: Longman.

Garside, Roger *et al.* 1993. Manual of information for the Lancaster Parsed Corpus. University of Lancaster. Unpublished.

Qiao, Hong Liang. Forthcoming. Corpus-trained parsing. Ph.D. dissertation. The University of Queensland.

Sampson, Geoffrey. 1992. The Susanne Corpus. University of Sussex. Unpublished paper.

Sampson, Geoffrey. 1993. The need for grammatical stocktaking. *Literary and Linguistic Computing*, 8:267–273.

Appendix A: THE SUSANNE FORMTAGS

Root-level formtags

O	paragraph
Oh	heading
Ot	title (e.g. of book)
Q	quotation

I	interpolation
Iq	tag question
Iu	scientific citation

Clause-level formtags

S	main clause
Ss	quoting clause embedded within quotation
Fa	adverbial clause
Fn	nominal clause
Fr	relative clause
Ff	“fused” relative
Fc	comparative clause
Tg	present participle clause
Ti	infinitival clause
Tn	past participle clause
Tf	*for-to* clause
Tb	“bare” nonfinite clause
Tq	infinitival relative clause
Z	reduced (“whiz-deleted”) relative clause
L	other verbless clause
A	special *as* clause
W	*with* clause

Phrase-level formtags

N	noun phrase
V	verb group
J	adjective phrase
R	adverb phrase
P	prepositional phrase
D	determiner phrase
M	numeral phrase
G	genitive phrase

The various phrase categories take lower-case subcategory symbols which can be combined in any meaningful combination (e.g. the verb group *must have been noticed* would be formtagged “Vcfp”). The phrase subcategories are:

Vo	operator section of verb group, when separated from

	remainder of V, e.g. by subject-auxiliary inversion
Vr	remainder of V from which Vo has been separated
Vm	V beginning with *am*
Va	V beginning with *are*
Vs	V beginning with *was*
Vz	V beginning with other 3rd-singular verb
Vw	V beginning with *were*
Vj	V beginning with *be*
Vd	V beginning with past tense
Vi	infinitival V
Vg	V beginning with present participle
Vn	V beginning with past participle
Vc	V beginning with modal
Vk	V containing emphatic DO
Ve	negative V
Vf	perfective V
Vu	progressive V
Vp	passive V
Vb	V ending with BE
Vx	V lacking main verb
Vt	catenative V

Nq	*wh-* N
Nv	*wh...ever* N
Ne	*I/me* head
Ny	*you* head
Ni	*it* head
Nj	adjective head
Nn	proper name
Nu	unit noun head
Na	marked as subject
No	marked as nonsubject
Ns	singular N
Np	plural N

Jq	*wh-* J
Jv	*wh...ever* J
Jx	measured absolute J
Jr	measured comparative J
Jh	postmodified J

Rq	*wh-* R
Rv	*wh...ever* R
Rx	measured absolute R
Rr	measured comparative R
Rs	adverb conducive to asyndeton
Rw	quasi-nominal adverb

Po	*of* phrase
Pb	*by* phrase
Pq	*wh-* P
Pv	*wh...ever* P

Dq	*wh-* D
Dv	*wh...ever* D
Ds	singular D
Dp	plural D

Ms	M headed by *one*

NON-ALPHANUMERIC FORMTAG SUFFIXES

Formtags may also contain non-alphanumeric symbols, including:

?	interrogative clause
*	imperative clause
%	subjunctive clause
!	exclamatory clause or other item
"	vocative item

Other non-alphanumeric symbols represent co-ordination structure. Under the SUSANNE scheme, second and subsequent conjuncts in a co-ordination are analysed as subordinate to the first conjunct; thus a co-ordination of the form:

chi, psi, and omega

(whatever the grammatical rank of the word-sequences chi, psi, etc.) would be assigned a structure of the form:

[chi, [psi], [and omega]]

The formtag of the entire co-ordination is determined by the properties of the first conjunct (except for singular/plural subcategories in the case of phrase categories to which these apply); the later conjuncts (which will often be transformationally reduced) have nodes of their own whose formtags mark them as "subordinate conjuncts". The following symbols relate to co-ordination (and apposition) structure:

+	subordinate conjunct introduced by conjunction
–	subordinate conjunct not introduced by conjunction
@	appositional element
&	co-ordinate structure acting as first conjunct within a higher co-ordination (marked in certain cases only)

Co-ordination is recognised as occurring between words as well as between higher-rank tagmas. Therefore nonterminal nodes may have formtags consisting of wordtags followed by co-ordination symbols, thus (using "WT" to stand for an arbitrary wordtag):

WT&	co-ordination of words
WT+	conjunct within word-level co-ordination that is introduced by a conjunction
WT-	conjunct within word-level co-ordination not introduced by a conjunction

(A word-level co-ordination always takes an ampersand on its formtag; phrase or clause co-ordinations do so only in very restricted circumstances.)

Also, certain sequences of orthographic words, in certain uses, are regarded as functioning grammatically as single words ("grammatical idioms"). For instance, *none the less* would normally be treated as a grammatical idiom, equivalent to an adverb (for which the wordtag is RR). In such cases, the nonterminal node dominating the sequence has a formtag consisting of an equals sign suffixed to the corresponding wordtag; and the individual words composing the grammatical idiom are not wordtagged in their own right, but receive tags with numerical suffixes reflecting their membership of an idiom. (The sequence *none* *the less* would be formtagged RR=, and the words *none*, *the*, and *less* in this context would be wordtagged RR31 RR32 RR33.)

Note that formtags of the forms WT& WT+ WT- WT= rank as word-level formtags for the purposes of determining tree structure as discussed above.

Appendix B: DETAILS OF THE LPC PARSING SCHEME

Sentence tags

Sq and Si

"Sq" means "a piece of direct quotation", normally an independent piece of language which occurs in fictional dialogue enclosed in quotetion marks. "Si" means "an interpolated sentence", i.e. a grammatically independent piece of language which is inserted (normally enclosed in brackets) in another sentence, but is not grammatically part of it. Note the following conventions used in handling direct quotations:

Pattern A:

"Nothing will change my mind", said Pat.

Pattern B:

Pat said, "Nothing will change my mind".

In these cases, the direct speech is analyzed as [Sq]:

Pattern A: [S "[Sq]" , [V] [N] S]
Pattern B: [S [N] [V] , "[Sq]" S]

Pattern C:

"Nothing," said Pat, "will change my mind".

In this case, Sq isn't used. Instead, the reporting clause is treated as an Si:

Pattern C: [S "..." , [Si] , "..." S]

Here is a further example of the use of Si:

That year ([Si how well I remember it! Si]) saw the beginning of my acting career.

S&, S+ and S-

"S&" represents a compound sentence, "S+" represents the second or subsequent conjoin of a compound sentence, if it begins with a coordinating conjunction, and "S-" represents such a conjoin when it does not begin with a coordinating conjunction.

Finite clause tags

F

A finite subordinate clause, i.e. a clause which contains a finite verb, and which is grammatically included in a sentence, is symbolized “F”. Typically, the “F” is followed by another symbol as detailed below.

Fa

“Fa” is a finite adverbial clause (e.g. a finite subordinate clause of time, of condition, of reason etc.)

E.g.: “[Fa Now that I have found out Fa] it may be easier for me to say it.“

Fc

“Fc” is a comparative clause, normally beginning with *than* or *as*.

E.g.: “He is cleverer [Fc than I thought Fc].”

Fn

“Fn” is a finite nominal clause, i.e. a finite subordinate clause which functions in the position of a noun phrase. Examples of “Fn” are *that*-clauses and *wh*-clauses (including indirect statements and indirect questions, also including “zero *that*-clauses”, where the that is omitted at the beginning of the clause).

E.g.: “I know [Fn that you saw them Fn].”

Fr

“Fr” is a relative clause, whether restrictive or non-restrictive,

E.g.: “the house [Fr in which I was born Fr]”

N.B. a “fused” or “nominal relative clause” as in “I will do [what you want]” is treated as “Fn”.

F&, F+, F-, etc.

These tags, which will also occur in combination with the letters ‘a’, ‘n’, ‘r’ etc., are used for coordinated finite subordinate clauses.

Non-finite and verbless clause tags

T
Nonfinite clauses are indicated by “T”. However, “T” does not normally occur alone. It is combined with the subscripts below.

Ti
“Ti” stands for a *to*-infinitive clause (e.g. an infinitive construction in which *to*+infinitive may or may not be followed by an object, a complement and/or adverbials).

E.g.: “It was a pity [Ti to leave them behind Ti].”

Tg
“Tg” stands for an *-ing* clause (i.e. a participial or gerundival construction in which the *-ing* form of the verb may or may not be followed by an object, a complement, and/or adverbials).

E.g.: “... where he first saw light machine guns [Tg being assembled Tg].”

Tn
“Tn” stands for a past participle clause (i.e. a construction in which the past participle form of the verb may or may not be followed by an object, a complement and/or adverbials).

E.g.: “[Tn Disappointed by the outcome Tn], John proceeded ...”

Tb
“Tb” stands for a “bare infinitive clause” (i.e. a construction in which the “bare infinitive”, infinitive without *to*, may or may not be followed by an object, a complement and/or abverbials).

E.g.: “We saw her [Tb cross the street hurriedly Tb].”

Tf
“Tf” is used as a variant of the infinitive clause, where the subject of the infinitive is introduced by *for*.

E.g.: “That would be a lot [Tf for them to swallow Tf].”

N.B. Nonfinite clauses generally have no subject: but it is also possible for a subject to occur;

E.g.: “I never yet heard of [Tg a young lady dying of love Tg]”.

W
“W” stands for a nonfinite or verbless clause introduced by *with*.

E.g.: “... another job [W with vastly more to offer W].”
“[W With RenÇ dying so unexpectedly W], we don’t know which way to turn.”
“He sauntered in [W with his hands in his pockets W].”

L
“L” stands for a verbless clause not introduced by *with* or by a subordinating conjunction.

E.g.: “[L Afraid of the consequences L], he hid the gun in a cupboard.”
“[L The Luger ready L], he walked simply back.”

Note: If an adverbial verbless clause or nonfinite clause is introduced by a subordinating conjunction (e.g. *if*, *when*), it is treated as a “Fa”:

E.g.: “The liner [Fa when finished Fa] will be the largest passenger vessel built in Europe since the war.”
“[Fa If in doubt Fa], leave the decision to your superior.”

If an adverbial verbless clause or nonfinite clause is introduced by a *wh*-word *why*, *what*, *how*, it is treated as a “Fn”:

E.g.: “We didn’t know [Fn what to do Fn].”
“They are leaving the village. Nobody knows [Fn why Fn].”

Constituent tags for major phrase types

V
“V” means “finite Verb Phrase”, in the narrow sense, in which “verb phrase” excludes objects, complements, etc. Thus “V” may include simple verb phrases such as *is*, *have, did* and also more complicated ones with modals, progressive aspect, perfective aspect or passive.

Vo and Vr

In general, no subscript is used with “V”. However, “Vo” and “Vr” are exceptions. They are used when a verb phrase is split into two parts by subject-auxiliary inversion. The first part is labelled “Vo” (o = “operator”) and the second part is labelled “Vr” (r = “remainder”). E.g. in “Have you seen Mary?” *have* is “Vo” and *seen* is “Vr”.

Note that “V” includes the negative word *not* as well as adverbs. E.g. the whole of *have not seen* (or *haven't seen* or *have recently seen*) is a “V”. But if the subject noun phrase occurs between the auxiliary and the main verb, this is treated as a separate noun phrase. Accordingly, *have you seen* consists of “Vo” followed by “N” followed by “Vr”.

Vi, Vg, Vn

These are labels for nonfinite verb phrases, i.e. verb phrases which are the verb phrases of nonfinite clauses “Ti”, “Tg” or “Tn”.

Vi

means “*to*-infinitive verb phrase”, e.g. *to eat* or *to have eaten*.

Vg

means “*-ing* participle verb phrase”, e.g. *eating* or *having eaten*.

Vn

means “past participle verb phrase”, e.g. *eaten*.

N

N is the label for a noun phrase, whether it is a single word (such as the pronoun *it*) or a sequence of words.

Na

In general, “N” has no subscripts. One major exception is “Na”, which stands for a noun phrase marked as subject of the verb. In practice, “Na” almost always indicated one of the pronouns *I*, *she*, *he*, *we*, *they*. (N.B. *you* and *it* as subject are not marked “Na” because their status of subject is not unambiguously shown by their form.)

Nq

Another exceptional use of “N” + subscript, meaning a *wh*-noun phrase, such as *who*, *which*, *which car*, *what time* etc.

J
"J" means an adjective phrase such as *happy*, *very tall*, *too happy for words*, etc. If an adjective occurs as the head of a noun phrase, e.g. *the wealthy*, *the unemployed*, the phrase is marked "N" not "J".

Jq
Here, as with "Nq", the "q" means "a phrase beginning with a *wh*-word", e.g. *How old*.

P
"P" stands for "prepositional phrase", e.g. *in London* or *on arriving at the station*, *with it*, *for what we are about to receive*, i.e. a preposition followed by its complement or completive element. Prepositional phrases also sometimes contain adverbs like *just* in *just inside the door*.

Pq
stands for "prepositional phrase with a *wh*-word, e.g. *on whose behalf, in which case, for whom*".

Po
stands for a "prepositional phrase beginning with the preposition *of*".

R
"R" is the symbol for an adverb phrase, which may be a single word such as *there* or *quickly* or may be a sequence such as *quite often*, *too fast, further than I expected,* etc.

Rq
stands for an adverb phrase beginning with a *wh*-word. This would include such phrases as *how* in *How do you feel*?, or *how long* in *How long have you been waiting*?

Constituent tags for minor phrase types

M
"M" stands for a "numeric phrase" when such an expression is part of a noun phrase. Examples are *five thousand* in *five thousand young people*; *another hundred* in *another hundred calories*. Numeric phrases have a numerical word as their head (e.g. *hundred*), and consist of at

least two words. (N.B. If numerical expressions such as *five thousand* occur on their own as noun phrases, they are labelled "N".)

D
"D" stands for a "determiner phrase", i.e. a phrase consisting of at least two words, in which the determiner is a head, and which is part of a noun phrase. E.g. *too many* in *too many people*; *a good few* in *a good few people*. (N.B. *too many* or *a good few* on their own, acting as a noun phrase, are labelled "N".)

Dq
stands for a determiner phrase (as defined above) beginning with a *wh*-word. E.g. *how many* and *how much*, when they are part of a noun phrase, as in *How many apples (did you buy)*?

G
"G" stands for "genitive phrase" i.e. a phrase which consists of two or more words acting as the genitive in a noun phrase. E.g. *the earth's* in *the earth's rotation around the sun*; *my mother's* in *my mother's greatest wish*; *last Friday's* in *last Friday's Evening Standard*; *someone else's* in *someone else's bedroom*; *the Vicar of Bray's* in *the Vicar of Bray's famous dictum.*

X
"X" is the negative word *not* when acting as an independent element of clause structure; e.g. in "He told us [Fn what not to do Fn]", *not* follows the object of the subordinate *what*-clause and precedes the verb phrase *to do*. Thus, the clause *what not to do* has the three constituents "Nq", "X", and "Vi". Generally, *not* is part of the verb phrase (see under "V" above) and therefore does not require an "X".

E
"E" is the label used for existential *there*, i.e. the unstressed *there* in the *there is/are* construction. E.g. "[E There E] is nothing wrong".

U
"U" is the tag used for an exclamatory word, such as *oh,* or a grammatical isolate, such as *yes* or *no*.

Appendix C: Results of the mapping between the parsing schemes of the Lancaster Parsed Corpus (LPC) and the Susanne Corpus

LPC	Susanne	Type	Solution
Sentence tags:			
S	S	1:2	S
Sq	Q	2:1+	Sq
Si	Ss	2:1+	Si
–	S@	2:2+	S-
–	S?	2:2	S
–	S*	2:2	S
–	S%	2:2	S
–	S!	2:2	S
Clause tags:			
Fa	Fa	1:1	Fa
Fc	Fc	1:1	Fc
Fn	Fn	1:2	Fn
Fr	Fr	1:1	Fr
–	Ff	2:2	Fn
Ti	Ti	1:1	Ti
Tg	Tg	1:1	Tg
Tn	Tn	1:1	Tn
Tb	Tb	1:1	Tb
Tf	Tf	1:1	Tf
–	Tq	2:2	Fr
W	W	1:1	W
L	L	1:1	W
–	A	2:2	Fa
–	Z	2:2	Fn
Phrase Tags:			
V	V	1:2	V
Vo	Vo	1:1	Vo
Vr	Vr	1:1	Vr

Vi	Vi	1:1	Vi
Vg	Vg	1:1	Vg
Vn	Vn	1:1	Vn
–	Vm	2:2	V
–	Va	2:2	V
–	Vs	2:2	V
–	Vz	2:2	V
–	Vw	2:2	V
–	Vj	2:2	V
–	Vd	2:2	V
–	Vc	2:2	V
–	Vk	2:2	V
–	Ve	2:2	V
–	Vf	2:2	V
–	Vu	2:2	V
–	Vp	2:2	V
–	Vb	2:2	V
–	Vx	2:2	V
–	Vt	2:2	V
N	N	1:2	N
Na	Na	1:1	Na
Nq	Nq	1:2	Nq
–	Nv	2:2	Nq
–	Ne	2:2	N
–	Ny	2:2	N
–	Ni	2:2	N
–	Nj	2:2	N
–	Nn	2:2	N
–	Nu	2:2	N
–	No	2:2	N
–	Ns	2:2	N
–	Np	2:2	N
–	N"	2:2	U
J	J	1:2	J
Jq	Jq	1:2	Jq
–	Jv	2:2	Jq
–	Jx	3:1+	{zero}
–	Jr	2:2	J
–	Jh	2:2	J
P	P	1:2	P
Pq	Pq	1:2	Pq
Po	Po	1:1	Po
–	Pb	2:2	P
–	Pv	2:2	Pq
R	R	1:2	R
Rq	Rq	1:2	Rq
–	Rv	2:2	Rq
–	Rx	2:2	R

–	Rr	2:2	R
–	Rs	2:2	R
–	Rw	2:2	R
M	M	1:2	M
–	Ms	2:3+	M, N
D	D	2:4+	D, {zero}, N
Dq	Dq	1:1	Dq
–	Dv	2:2	Nq
–	Ds	2:4+	D, N, R
–	Dp	2:4+	D, N
G	G	1:1	G
X	–	4:1+	X
E	–	5:1+	E
U	–	5:1+	U

"+" means that a small program is needed to process the mapping or a manual mapping is needed.
"–" means that there is no such parsing tag in that parsing scheme.
{zero} means empty or nothing.