SYLVIANE CARDEY
PETER GREENFIELD
Centre de recherche en linguistique et traitement automatique des langues Lucien Tesnière, Université de Franche-Comté, Besançon cedex, France
sylviane.cardey@univ-fcomte.fr, peter.greenfield@univ-fcomte.fr

Disambiguating and Tagging Using Systemic Grammar

Abstract

We present a compositional modelling technique with an associated computationally effective calculus, Systemic Grammar, which has as intention the exhaustive modelling and exploitation of linguistic phenomena observed in and between languages. The models so produced are self-explanatory and errors made in their construction are thus easy to identify. As an application we describe the Labelgram system, showing how, by using our methodology, automatic morphological disambiguated tagging of text can be achieved, with illustrations drawn from English, French, German and Spanish. We discuss an application of Labelgram, that of establishing a comprehensive benchmark for Labelgram itself.

Keywords: Computational Linguistics, Disambiguating Tagger, Grammatical Text Encoding, Labelgram, Linguistic Quality, Microsystem, Neologisms, Systemic Grammar.

1. Introduction

Natural language is complex. However, though languages are open systems, linguistics advances by the exhaustive study of linguistic phenomena. Fortunately, languages seem to display regularities which need to be brought into prominence. Thus modelling techniques for natural language should not only admit this complexity with regularities as well as the open nature of language, but should also simplify our understanding of language. A model should serve not only as a description but also as a resource for solving other linguistic problems covered by the model as well as linguistic problems not yet perceived in which the model and perhaps others are involved.

Systemic Grammar [Cardey & Greenfield 2002] incorporates techniques of expression that are conducive to the exhaustive analytical and compositional modelling and subsequent exploitation of linguistic phenomena observed in and between languages. The representations so produced are economical being intentionally based (marrying set enumeration and comprehension) as well as self-explanatory, and errors made in their construction are thus easy to identify. Using abstract interpretation techniques, the well formedness of representations can be verified automatically, and properties such as complexity and quality measures can be calculated. Furthermore, compile-time and run-time optimisation for particular speed performance sensitive applications can be effected. Applications are not limited to 'end-user' tools but are also used by design and necessity by the linguist as research tools in extending a given language system's grammar. Systemic Grammar is completely independent of the traditional 'divisions' of linguistics such as phonetics, lexis, morphology, syntax, morpho-syntax, semantics etc [Cardey & Greenfield 1997-2]. This is not to say that the linguist in constructing a model does not exploit his or her competence in these 'divisions' in expressing the representation of observed linguistic phenomena. On the contrary, he or she must necessarily exploit these as and when their need is put into evidence during analysis. As an application we describe the Labelgram system, showing how, by using our methodology, automatic morphological disambiguated tagging of text can be achieved, with illustrations drawn from English, French, German and Spanish. We finish in discussing Systemic Grammar's contribution to systemic quality modelling by means of an application which has as goal the establishment of a comprehensive benchmark for Labelgram itself.

2. Systemic Grammar

Systemic Grammar is based on the postulate that a language or indeed languages (as in translation) can be segmented into individual microsystems based on the observation that such microsystems influence each other.

> 'A microsystem is a system that is sufficiently small so as to be humanly prehensile and which can subsequently be processed by machine, but also sufficiently large so as to function formally as a whole. A microsystem is always a component of a whole to which the microsystem's properties can be extended and it is also possible to create links between the microsystems of a system of which they are part in order to obtain a complete formal description of the system constituted of microsystems that are nested, that have common parts or which function in identical and identifiable manners. All levels of language analysis can be so described (lexis, syntax, morpho-syntax, semantics, morpho-semantics and others).'
>
> Translation of the definition of Yves Gentilhomme [Gentilhomme 1985] adapted by Sylviane Cardey [Cardey 1987]

Each constituent microsystem of a language system comprises an invariant part and variant parts. What thus has to be done is to describe for each constituent microsystem its invariant part, which allows us to find and indeed discover the microsystem, to name it and to give it prominence as a system in its own right, as well as being a component part of the language system and within the language system. The variant parts of a microsystem need also to be described, these parts being due to the microsystem being put into relation with other microsystems of the language system, this in the knowledge that a system can be related with several systems which in their turn influence one another. Thus to begin with we have a recognisable 'invariant' system which is necessarily canonical in form and content and other systems each represented by a specific variant part. However, we are missing the system which relates the invariant part and the variant parts, this being thus a system of relationships.

It is this system of relationships which allows us to organise the microsystem in terms of its invariant canonical system and the variant part systems together with the relationships that then link the two.

2.1 Computational modelling

For the computational modelling of particular relationship pairs, themselves microsystems, we use a model and proof theoretic based approach [Greenfield 2003]. This enables us to ensure that we can have an explicitly exhaustive and computationally effective treatment in which each observed case is supported with evidence (such as corpus justifications), together with the merits of algebraic manipulation.

Given a relationship between two linguistic systems S and T, when we wish to find a value in T given a value in S we call S the source system and T the target system. In analytical computational applications, such as disambiguating taggers, S is in general variant and T invariant, whilst in generation typically the inverse holds. There can be intermediary applications in which both systems are invariant, and applications like translation where both systems are variant. Rather than employ an arbitrary binary relation, we use functions with the constraint that they be bijections, written S >–» T. That is, the function is applicative (total - the domain covers S), surjective (onto - the codomain or range covers T) and injective (one-to-one, each observed case is represented with both a domain value and also a codomain value. For quality justification, when S is a variant system, then each value in T incorporates an ancillary corpus example from S, and vice-versa. Being injective ensures both that our corpus is adequate at least in terms of there being a corpus example for each linguistic case, and also that the relationship has the advantage of also being a function in the inverse direction (source becomes target, and vice versa).

The technique that we use to incarnate a given base (not composite) relationship for some linguistic problem involves predicating over S and T by means of nested sets [Knuth 1975, pp.309, 314], and this results in the partitioning of the relationship, where each source element of the partition is the set of source data corresponding to a particular linguistic case. The cardinality of the partition is the number of linguistic cases. This process of predication is initially the analytical work of the linguist who, in applying his or her skill and devising and employing tools where necessary (see for example [Morgadinho & Venet 2003]), endeavours to ensure that, for automatic applications to concrete input data, the predication can be subsequently achieved by the machine alone. The resulting bijection is a totally ordered relation over the predicates $<$ and $\subset$, where $<$ is for the partitioning of immediate disjoint subsets of some nesting level and $\subset$ is for subsetting. $<$ and $\subset$ encompass then the particular predicate(s) discovered or devised by the linguist. Whilst $\subset$ is a linguistic predicate, $<$ is typically some standard but arbitrary ordering (this is so for the technique for raw tagging based on word endings discussed later in the paper). By arbitrary we mean linguistically arbitrary in the sense that alphabetic ordering, for example, is arbitrary and not a property of a language *per-se* but is standard as it is used as a linguistic tool for effecting and controlling standard normal forms (e.g. alphabetic ordering of tag set names). The total ordering aids varied processing of the linguist's representation. For example there is the possibility of firstly generating the model from the linguist's representation (expliciting set differencing due to $<$ and $\subset$) and then calculating from this more efficient or application effective functionally equivalent algorithms. In effect this means finding a $<$ which is more appropriate, drawing on for example decision table techniques [Dial 1970], [Humby 1973]. As for $<$ and $\subset$, in the case of automatic applications we can exploit these at compile time not only to verify that the linguist's representation of the relationship is well formed, but also for optimisation (such as eliminating repeated constraints in nested constraints across $\subset$ nesting sequences). Standardised total ordering also aids in computationally effected algebraic operations where the operands are themselves microsystems.

Microsystems as bijections are represented and thus defined either in the concrete base form by means of nested context rules (which have an algorithmic reading) or in the form of operations on such bijections (such as functional composition). The relationship between S and T can be viewed as a concordance for the specific linguistic phenomenon treated. In the base nested context rules representation, each rule embodies a context value in S, the corresponding value in T, and an indicator in the context giving the structural link with the T value.

3. Labelgram

Labelgram is a system for the automatic morphological disambiguated tagging of texts [Cardey *et al.* 1997], [Cardey & Greenfield 1997-1]. A prototype implementation exists for French texts [El Harouchy 1997] and work is in progress for English [Birocheau 2000], and for German and Spanish [Morgadinho & Venet 2003].

3.1 Labelgram and Systemic Grammar

In Labelgram, the overall systemic grammars are 'super' microsystems modelling the relationship between texts and morphologically disambiguated tagged texts. The source and variant system is the language (in the form of texts), the target and invariant system is the morphological disambiguated tagging of each text unit, and the relationship between the two is the (composite) system of linguistic data which implements this relationship.

3.2 Context rules

In the representation of a base microsystem relationship, a context rule consists of a description of linguistic units which is so organised so as to have a specific linguistic categorisation (of the type desired by the linguist). The nested form of context rules, each being either a set comprehension for regularities (defaults) or an enumeration for exceptions, results in an intentionally driven approach leading not only to representations which are efficient in terms of size [Cardey *et al.* 1997] but which accept for example, in the case of word dictionaries, neologisms 'obeing the rules for French word formation'. For example in Labelgram's French raw grammatical category tagger, the context rule which recognises and tags word forms ending in at least:

-er (1)

has 579 entries whilst the electronic Robert dictionary of French, an extensional dictionary, has 6666 entries. The overall default context tags (perhaps new) words in **-er** as Verbe inf. (verb infinitives).

3.3 Texts in Labelgram

In Labelgram, the text under analysis is in the form of sequences of 'units' where a unit, depending on the analysis, may be automatically annotated (e.g. tagged with grammatical category(ies)). For Labelgram in its current (including research) state, units include:

- individual letters where the 'text' is a 'word'; used for morphological phenomena in isolation, e.g. finding grammatical category(ies) as a function of the word's ending, as in the **-er** example (1) above.
- individual 'word' forms or compounds (treated as single entities); e.g. for grammatical category disambiguation as a function of tagged word forms/compounds in context with neighbouring units. Punctuation, text and sentence boundaries are also so classified.
- 'compounds'; a compound is treated as a single entity, to be tagged (before any disambiguation) with one or more grammatical categories. For example, for French, in:

  Il est arrivé **juste à temps** (2)
  (he arrived just in time)

  the machine can calculate that **juste à temps** is a compound, in fact an adverbial locution, that is with the function of an adverb [Petetin 2002]. Compounds can themselves also be polycatgorial. English compounds of the type preposition+determiner+noun are usually either adjectives or adverbs. Consider however

  on the nose (3)

  which can be either an adjective ('annoying') or an adverb ('precisely') [Cella 2001].

3.4 Disambiguating and Tagging

We start with a German example due to Anne-Laure Venet for the morphological ambiguity 'Appenzeller':

Der **Appenzeller** Bezirk entwickelt seine Industrie. (4)
(The Appenzell region develops its industry.)
Die **Appenzeller** haben den Ruf entgegenkommend sein. (5)
(The people of Appenzell have the reputation of being welcoming.)

In (4) **Appenzeller** is an adjective and in (5) a substantive. Raw tagging is based on a microsystem comprising a combination of closed sets (determiners, pronouns, conjunctions, interjections etc.) and the technique described above for French based on word form endings (example 1). *Bezirk* is classified as unicategorial {Subst.} and **Appenzeller** polycategorial {Adj., Subst.}. For polycategorial word forms recourse is made to another microsystem in which contexts are over sequences of elements in the sentence. For resolving the ambiguity on **Appenzeller** we have:

1. AF(-ler) –> Subst. : 'Die **Appenzeller** haben den Ruf entgegenkommend sein.'
   1.1 set of exceptions in the form of (AF(WORD_FORM) –> CATEGORY : CORPUS_EG)
   1.2 AF(^–) + SUBST –> Adj. : 'Der **Appenzeller** Bezirk entwickelt seine Industrie.'

in which 1.1 and 1.2 are order dependent (effecting set difference, the set of exceptions in 1.1 are 'subtracted' from the 'view' of 1.2). AF indicates the ambiguous word form (here **Appenzeller**), –ler indicates a word form ending in *ler* and ^– indicates any word form starting with a capital letter. In German a substantive never follows a substantive (except if separated by a punctuation sign which is not so here). A word starting with a capital letter followed by a substantive is an adjective. For (4), **Appenzeller** will be tagged as an adjective by context rule 1.2 (having passed 1., and 1.1 having failed), and for (5) as a substantive by the 'default' context rule 1., neither sub context rule 1.1 nor 1.2 being applicable.

As will be seen, context rules themselves operate in context with other context rules, the context rules are nested, and there are no explicit negations. Their appearance as a whole resembles a concordance, not of text but of grammatical contexts (in S) where each item being concorded corresponds to a category (in T), the item's structural position in the grammatical context being indicated by a marker whose name itself indicates the linguistic phenomenon being concorded (here 'AF'), and all this in the 'context' of marked (e.g. **Appenzeller**) corpus examples. Compile time verification of well formedness can be effected to ensure that for example in the set of exceptions 1.1 each and all the word forms do end in *ler* (proper subset of 1.). Also, 1.1 can be optimised so that in their compiled form, the word forms have their *ler* endings removed (redundant constraint having passed the default context constraint in 1.).

In the following Spanish example due to Helena Morgadinho:

El **almuerzo** en **este** restaurante fue delicioso. (6)
(The lunch in this restaurant was delicious.)

out of context, **almuerzo** and **este** are polycategorial. Raw tagging (similar to French and German) produces:

Word form	El	almuerzo	en	este	restaurante	fue	delicioso
Categories	{Art.}	{Sust., Vbo. conj.}	{Prep.}	{Sust., Adj., Adj. demons.}	{Sust.}	{Vbo. conj.}	{Adj.}

Disambiguation starts with {Sust., Adj., Adj. demons.} (**este**). Knowing that although *en* can be followed by several grammatical categories, it is also the case that concerning this type of ambiguity:

- *en* cannot be followed by an Adj., thus one category less;
- *en* can be followed by a Sust., but this is not possible with another Sust. following (*restaurante*), (at least a separating comma would be needed which is not the case here).

From this, the following context rule can be formulated, where AF indicates the ambiguous form (here **este**):

{en} + AF + SUST –> Adj. demons. : 'El almuerzo en **este** restaurante fue delicioso.'

Disambiguation is completed by resolving {Sust., Vbo. conj.} (**almuerzo**). Now, various grammatical categories can be found before a Vbo. conj., even punctuation signs, but never articles, so this leaves just the one possibility for **almuerzo**, Sust, giving the following context rule:

ART + AF –> Sust. : 'El **almuerzo** en este restaurante fue delicioso.'

The disambiguated tagging is thus:

Word form	El	almuerzo	en	este	restaurante	fue	delicioso
Category	{Art.}	{Sust.}	{Prep.}	{Adj demons.}	{Sust.}	{Vbo conj.}	{Adj.}

The following English example due to Gaëlle Birocheau:

The man who works here lives **in** London. (7)

with raw tagging similar to the previous languages, followed by disambiguating context rules gives:

Word form	The	man	who	works	here	lives	in	London	.
Raw categories	{DETsing, DETplu}	{N, V}	{PROinterr, PROrel}	{Nplu, V3}	{ADV}	{Nplu, V3}	{PREP, ADV, ADJ, N}	{N}	{PUNCT}
Results of Disambig-uation	{DETsing}	{N}	{PROrel}	{V3}	{ADV}	{V3}	{PREP, ADV, ADJ, N}	{N}	{PUNCT}

For **in**, in the absence of context rules dealing with composition or verbal complements, the four combinations

V+**PREP**+N, V+**ADV**+N, V+**ADJ**+N, V+**N**+N,

are all valid.

In the following French sentence due to Zahra El Harouchy:

la méchante rigole car le petit est malade (8)

(the nasty woman laughs because the little boy is ill)

all the word forms are polycategorial and the raising of the ambiguities is initially due to resolving {Nom,Conj.} on the word *car*. For this sentence as input, the prototype Labelgram implementation gives the following output:

Word form	Tagger ref.	Categories	Disambig n°	Disambig ref.	Category
la	2.12/	{Art, Nom, Pro pers.}	5	45/	{Art.}
méchante	41.2/	{Nom, Adj.}	28	339/	{Nom}
rigole	360.4/	{Nom, Verbe conj.}	8	79/	{Verbe conj.}
car	47.5/	{Nom, Conj.}	10	144/	{Conj.}
le	pre_dict	{Art., Pro. pers.}	5	45/	{Art.}
petit	279.1/	{Nom, Adj.}	28	339/	{Nom}
est	pre_dict	{Nom, Verbe conj.}	8	74/	{Verbe conj.}
malade	13.1/	{Nom, Adj.}	28	346a/	{Adj.}

3.5 Labelgram functional architecture

The functional architecture for Labelgram for French for simple words can be described as follows (using the notation of the formal specification language Z [Spivey 1992]). Consider the following microsystem declarations:

$\mu_{segmenter}$: *Text* >–» *Segmented_Text*

μ_{tagger} : *Segmented_Text* >–» *Tagged_Text*

$\mu_{disambiguator}$: *Tagged_Text* >–» *Disambiguated_Tagged_Text*

In fact $\mu_{disambiguator}$ is a union of a set of specific disambiguators, each for a distinct family of polycategorial ambiguity cases:

$\mu_{disambiguator} = \cup \{ \mu_{disambiguator_polycat_1}, \mu_{disambiguator_polycat_2}, \ldots \}$

The sets of tags in the raw tagged text are related to the appropriate polycategorial disambiguator with the following notional bijectional microsystem declaration:

$\mu_{tagset_disambiguator_polycat_i}$: IP*TagSet* >–» $\mu_{disambiguator_polycat_i}$

where IP is the power set operator. Note that for $\mu_{tagset_disambiguator_polycat_i}$ both source S and target T are invariant.

Using forward functional composition (the infix operator ;), we obtain the complete disambiguating tagger microsystem:

$\mu_{disambiguating_tagger}$: *Text* >–» *Disambiguated_Tagged_Text*

$\mu_{disambiguating_tagger} = \mu_{segmenter} ; \mu_{tagger} ; \mu_{disambiguator}$

The disambiguated tagging example above amounts to the evaluation of the following function application:

$\mu_{disambiguating_tagger}$ ('la méchante rigole car le petit est malade')

4 Benchmarking

A particular bugbear in Human Language Technology concerns the provision of static tracability and thus comprehensive benchmarks which are of adequate quality. Because a systemic grammar is case based, it should

be comprehensive (correct and complete), and thus for it the word 'adequate' has a clear meaning. The point here is that because Labelgram is based on Systemic Grammar, a benchmark corpus is intrinsic to Labelgram in its linguistically developed and validated form. Now, Labelgram can be used itself as a tool to construct its own benchmark corpus with the particularity that such a corpus is either representatively exhaustive or if not exhaustive one at least knows what is missing from the corpus. The Labelgram implementation includes dynamic linguistic tracing; not only the recording of which linguistic rules are successful in disambiguating (see the output table above) but a trace of the rules visited until a successful solution is found.

It is incumbent on the linguist to provide for each case of polycategorial ambiguity an example text exhibiting and identifying this ambiguity (for example (8), case '144/' above in which 'car' is the identified ambiguous item (nom (noun) or conjonction (conjuction)). Such a text is either preferably a corpus and thus performance based text (together with its provenance) or, in the lack of this, a synthetic text devised by the linguist and depending thus on his or her competence. During the application of Labelgram to real texts, when a polycategorial ambiguity is encountered for which there is only a synthetic example, the system has the possibility to add the acquired real example as corpus based together with its provenance (this procedure being under the control of the linguist who must ensure the validity and efficacy of the 'proposed' real example), thus enriching the corpus based content of the test data set in the direction of a benchmark corpus. The component microsystems are treated in the same way.

5. Conclusion

Systemic Grammar enables linguists to express themselves in a natural way but in a way that can be exploited by the machine not only for specific applications but also as a research tool. In its practice, linguistic phenomena have been explored and categorised and microsystems have been constructed from others (e.g. the automatic generation of German determiners with material from the microsystem for declining German adjectives [Cardey & Greenfield 2000]). Systemic Grammar marries linguistic performance and linguistic competence and this in an exhaustive manner, to the extent that it entails an intrinsic systemic quality approach enabling the production of comprehensive benchmarks.

References

[Birocheau 2000] Birocheau, G., "Morphological Tagging to Resolve Morphological Ambiguities". In Proceedings of the Second International Conference on Language Resources and Evaluation (LREC2000) 2000.

[Cardey 1987] Cardey S.: Traitement algorithmique de la grammaire normative du français pour une utilisation automatique et didatique, Thèse de Doctorat d'Etat, Université de Franche-Comté, France, June 1987

[Cardey & Greenfield 1997-1] Cardey, S., Greenfield, P.G., "Ambiguïté et traitement automatique des langues. Que peut faire l'ordinateur ?". In Actes du 16è Congrès International des Linguistes, Paris, 20-25 July 1997. Elsevier Sciences (CD ROM).

[Cardey & Greenfield 1997-2] Cardey, S., Greenfield, P.G., "Peut-on séparer lexique, syntaxe, sémantique en traitement automatique des langues ?". In Cahiers de lexicologie 71 1997, pp 37-51, ISSN 007-9871.

[Cardey *et al.* 1997] Cardey, S., El Harouchy, Z., Greenfield, P.G., "La forme des mots nous renseigne-t-elle sur leur nature ?". In Actes des 5èmes journées scientifiques, Réseau Lexicologie, Terminologie, Traduction, LA MEMOIRE DES MOTS, Tunis, 22-23 September 1997. AUPELF-UREF. pp 305-313.

[Cardey & Greenfield 2000] Cardey, S., Greenfield, P.G., "Génération automatique de déterminants et d'informations sur les déterminants en allemand en contexte". In Actes du Colloque DET2000 – Détermination et Formalisation, Bellaterra, Université Autonome de Barcelone, 25-26 February 2000.

[Cardey & Greenfield 2002] Cardey, S., Greenfield, P.G., "Systemic Language Analysis and Processing", paper accepted. Second Conference of the FNLP, Language, Brain and Computation: Venice, October 3-5, 2002.

[Cella 2001] Cella, L., "Vers une Reconnaissance Automatique des Composés Anglais à Trois Eléments et de leur Catégorie Syntaxique". Mémoire de Maîtrise, mention : Industries de la Langue, Univesité de Franche-Comté, Besançon, France, 2001.

[Dial 1970] Dial, R.B., "Algorithm 394. Decision Table Translation", Communications of the ACM, Vol. 12, No. 9, September 1970, pp 571-572.

[El Harouchy 1997] El Harouchy, Z., "Dictionnaire Automatique et Grammaire pour le traitement automatique des ambiguïtés morphologiques des mots simples en français". In: Actes du Colloque International FRACTAL, Besançon, France, ISSN 0758 6787, 10-12 décembre, 1997, pp 141-151.

[Gentilhomme 1985] Gentilhomme, Y., "Essai d'approche microsystémique, Théorie et pratique, Application dans le domaine des sciences du langage". Peter Lang, 1985.

[Greenfield 2003] Greenfield, P., "A microsystemic computational model for natural language processing". To appear in "Microsystèmes et TA", special number of the BULAG, 2003.

[Humby 1973] Humby, E., "Programs from decision tables", Macdonald/American Elsevier, ISBN 0 356 04126 3/ISBN 0 444 19569 6, 1973.

[Knuth 1975] Knuth, D., "The art of computer programming", Second edition, Volume 1 / Fundamental Algorithms, Addison-Wesley Publishing Company, Reading, Massachusetts, 1975.

[Morgadinho & Venet 2003] Morgadinho, H., Venet, A.-L., "Construction de dictionnaires microsystèmiques". To appear in "Microsystèmes et TA", special number of the BULAG, 2003.

[Petetin 2002] Petetin, L., "Reconnaissance Automatique des Composés à Trois Eléments : Ambiguïté Morphologique et Grammaticale". Mémoire de Master, mention : Industries de la Langue, Université de Franche-Comté, 2002.

[Spivey 1992] Spivey, J.M., "The Z Notation", Prentice Hall,1992.