Journal of Language Modelling

VOLUME 1 ISSUE 1
JUNE 2013

Institute of Computer Science
Polish Academy of Sciences
Warsaw

Journal of Language Modelling

VOLUME 1 ISSUE 1
JUNE 2013

JOURNAL OF
LANGUAGE MODELLING

ISSN 2299-8470 (electronic version)
ISSN 2299-856X (printed version)
http://jlm.ipipan.waw.pl/

MANAGING EDITOR
Adam Przepiórkowski IPI PAN

SECTION EDITORS
Elżbieta Hajnicz IPI PAN
Agnieszka Mykowiecka IPI PAN
Marcin Woliński IPI PAN

STATISTICS EDITOR
Łukasz Dębowski IPI PAN

Published by IPI PAN
Instytut Podstaw Informatyki
Polskiej Akademii Nauk
Institute of Computer Science
Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland

Layout designed by Adam Twardoch.
Typeset in XeLaTeX using the typefaces: *Playfair Display*
by Claus Eggers Sørensen, *Charis SIL* by SIL International,
JLM monogram by Łukasz Dziedzic.

Except where noted, all content is licensed under
the Creative Commons Attribution 3.0 Unported License.
http://creativecommons.org/licenses/by/3.0/

EDITORIAL BOARD

Steven Abney University of Michigan, USA

Ash Asudeh Carleton University, CANADA;
University of Oxford, UNITED KINGDOM

Chris Biemann Technische Universität Darmstadt, GERMANY

Igor Boguslavsky Technical University of Madrid, SPAIN;
Institute for Information Transmission Problems,
Russian Academy of Sciences, Moscow, RUSSIA

António Branco University of Lisbon, PORTUGAL

David Chiang University of Southern California, Los Angeles, USA

Greville Corbett University of Surrey, UNITED KINGDOM

Dan Cristea University of Iași, ROMANIA

Jan Daciuk Gdańsk University of Technology, POLAND

Mary Dalrymple University of Oxford, UNITED KINGDOM

Darja Fišer University of Ljubljana, SLOVENIA

Anette Frank Universität Heidelberg, GERMANY

Claire Gardent CNRS/LORIA, Nancy, FRANCE

Jonathan Ginzburg Université Paris-Diderot, FRANCE

Stefan Th. Gries University of California, Santa Barbara, USA

Heiki-Jaan Kaalep University of Tartu, ESTONIA

Laura Kallmeyer Heinrich-Heine-Universität Düsseldorf, GERMANY

Jong-Bok Kim Kyung Hee University, Seoul, KOREA

Kimmo Koskenniemi University of Helsinki, FINLAND

Jonas Kuhn Universität Stuttgart, GERMANY

Alessandro Lenci University of Pisa, ITALY

Ján Mačutek Comenius University in Bratislava, SLOVAKIA

Igor Mel'čuk University of Montreal, CANADA

Glyn Morrill Technical University of Catalonia, Barcelona, SPAIN

Reinhard Muskens Tilburg University, NETHERLANDS

Mark-Jan Nederhof University of St Andrews, UNITED KINGDOM

Petya Osenova Sofia University, BULGARIA

David Pesetsky Massachusetts Institute of Technology, USA

Maciej Piasecki Wrocław University of Technology, POLAND

Christopher Potts Stanford University, USA

Louisa Sadler University of Essex, UNITED KINGDOM

Ivan A. Sag Stanford University, USA

Agata Savary Université François Rabelais Tours, FRANCE

Sabine Schulte im Walde Universität Stuttgart, GERMANY

Stuart M. Shieber Harvard University, USA

Mark Steedman University of Edinburgh, UNITED KINGDOM

Stan Szpakowicz School of Electrical Engineering
and Computer Science, University of Ottawa, CANADA;
Institute of Computer Science,
Polish Academy of Sciences, Warsaw, POLAND

Shravan Vasishth Universität Potsdam, GERMANY

Zygmunt Vetulani Adam Mickiewicz University, Poznań, POLAND

Aline Villavicencio Federal University of Rio Grande do Sul,
Porto Alegre, BRAZIL

Veronika Vincze University of Szeged, HUNGARY

Yorick Wilks Florida Institute of Human and Machine Cognition, USA

Shuly Wintner University of Haifa, ISRAEL

Zdeněk Žabokrtský Charles University in Prague, CZECH REPUBLIC

Constructions with Lexical Integrity*

Ash Asudeh[1,2], *Mary Dalrymple*[2], *and Ida Toivonen*[1]
[1] Carleton University
[2] Oxford University

ABSTRACT

Construction Grammar holds that unpredictable form-meaning combinations are not restricted in size. In particular, there may be phrases that have particular meanings that are not predictable from the words that they contain, but which are nonetheless not purely idiosyncratic. In addressing this observation, some construction grammarians have not only weakened the word/phrase distinction, but also denied the lexicon/grammar distinction. In this paper, we consider the word/phrase and lexicon/grammar distinction in light of Lexical-Functional Grammar and its Lexical Integrity Principle. We show that it is not necessary to remove the word/phrase distinction or the lexicon/grammar distinction to capture constructional effects, although we agree that there are important generalizations involving constructions of all sizes that must be captured at both syntactic and semantic levels. We use LFG's *templates*, bundles of grammatical descriptions, to factor out grammatical information in such a way that it can be

Keywords: *syntax, lexicon, semantics, constructions, Lexical Integrity, templates, Swedish, Dutch, LFG, Glue Semantics*

*The authors contributed equally and are listed in purely alphabetical order. This paper is a revised and expanded version of Asudeh *et al.* (2008). We gratefully acknowledge the support of the following grants: an Early Researcher Award from the Province of Ontario (Asudeh), Leverhulme Research Fellowship #RF-2012-295 (Dalrymple), NSERC Individual Discovery Grant #371969 (Asudeh), and SSHRC Standard Research Grant #410-2010-1841 (Toivonen). For helpful comments, we thank the audience at LFG08, in particular Kersti Börjars, Marie-Elaine van Egmond, Dag Haug, Helge Lødrup, and Nigel Vincent. We also thank the audience at SE-LFG 5. For comments on earlier drafts of this paper, we are grateful to Elizabeth Christie, Ray Jackendoff, Stefan Müller, and Ivan Sag. Any remaining errors are our own. Lastly, we thank the editors, staff, and reviewers at *JLM*.

invoked either by words or by construction-specific phrase structure rules. Phrase structure rules that invoke specific templates are thus the equivalent of phrasal constructions in our approach, but Lexical Integrity and the separation of word and phrase are preserved. Constructional effects are captured by systematically allowing words and phrases to contribute comparable information to LFG's level of functional structure; this is just a generalization of LFG's usual assumption that "morphology competes with syntax" (Bresnan, 2001).

1 WORDS, CONSTRUCTIONS, AND THE LEXICON

The observation that unpredictable form-meaning combinations are not restricted in size forms the basis for Construction Grammar (Fillmore, 1988; Fillmore *et al.*, 1988; Kay and Fillmore, 1999; Goldberg, 1995, 2006; Goldberg and Jackendoff, 2004; Michaelis, 2010; Sag, 2010; Boas and Sag, 2012).[1] A meaning that is associated with a word or a part of a word may also be associated with a phrasal structure in the same language, or in another language. Construction Grammar takes the structure and organization of the collection of listemes/constructions as crucially important, and a central concern is the study of the relations among constructions: this contrasts with Di Sciullo and Williams (1987), who consider the collection of listemes to be unstructured and the study of the relations among listemes uninteresting. Examples such as (1)–(3) involve correspondences between phrasal structures and idiosyncratic meanings; the syntactic frame of the multi-word expression itself, perhaps along with some specifications on what words are permitted, evokes some interpretation.

(1) The bigger the better. (Fillmore *et al.*, 1988; Culicover and Jackendoff, 1999)

(2) What's that koala doing sleeping in the corner? (Kay and Fillmore, 1999)

(3) Smithy drank his way through university. (Jackendoff, 1990; Goldberg, 1995)

[1] See Sag *et al.* (2012) for a historical overview of Construction Grammar and further references. See Sag (2012) for an informal overview of a formal theory of Construction Grammar (Sign-Based Construction Grammar).

Most words in the expressions above are exchangeable for other words, so they seem more flexible than prototypical idioms. Yet their form and associated interpretation must be learned by English speakers, as these constructions do not, it is argued, follow from general compositional principles of English grammar. On the Construction Grammar view, such expressions are not a peripheral part of the grammar which need not be accounted for in grammatical theory; instead, they lie at one end of a continuum of structures relating forms to meanings. Constructions as pairings of form and meaning can be larger or smaller than a word, and can have structure and meaning that is general or predictable to a variable extent. Following on from this view, Construction Grammarians have denied the utility of a strict division between word-internal grammatical regularities and phrasal regularities, or between semantically bleached grammatical structures and structures which contribute specialized or irregular meanings. As a consequence, Construction Grammarians have tended to emphasize commonalities across types of constructions rather than differences, and to de-emphasize differences between word-level and phrase-level constructions.

A distinction between words and multi-word expressions is not fundamental to Construction Grammar, although such a distinction is not necessarily in principle ruled out. In fact, some advocates of Construction Grammar go so far as to claim that the distinction between lexicon and grammar is no longer useful, as illustrated by the following quotes:

> [M]orphemes are clear instances of constructions in that they are pairings of meaning and form that are not predictable from anything else. It is a consequence of this definition that the lexicon is not neatly differentiated from the rest of grammar. (Goldberg, 1995, p. 4)

> In Construction Grammar, no strict division is assumed between the lexicon and syntax. (Goldberg, 1995, p. 7)

> Every theory of language has to take a word to be a complex of phonological, syntactic, and semantic structures; commonly, the store of words is called the *lexicon*. … *Aspects* [(Chomsky, 1965)] treats the lexicon as a component of language distinct from the rules of grammar. Words are taken to

> be the locus of irregularity in language, while rules of grammar encode all the regularities. Words get into sentences by being inserted into syntactic derivations, at the point when syntactic trees are being built and before trees begin to be manipulated and fed to phonology and semantics. … But while [this view of the lexicon] was altogether plausible in the context of early work in generative grammar, I believe that subsequent developments reveal it as another major mistake that has remained in the background as unquestionable dogma within the mainstream school of thought. (Jackendoff, 2007, p. 53)

It is also telling that the section that contains this last quote is titled "Another Fundamental Mistake: The Lexicon/Grammar Distinction".

Another perspective on this fundamental issue arises if we ask whether there is any necessary theoretical distinction between words and phrases. Many linguists agree that words and phrases must be distinguished (Anderson, 1992; Aronoff, 1993; Sadler and Spencer, 2000; Stump, 2001; Blevins, 2006), though the word/phrase distinction is denied by some linguists, not just some working in Construction Grammar, but also those working within the Distributed Morphology paradigm (Halle and Marantz, 1993, 1994; Marantz, 1997; Siddiqi, 2009), which otherwise has quite different morphosyntactic assumptions.

We believe that there is a fundamental reason to maintain the distinction between words and phrases that has previously gone unaddressed in the literature. Natural language morphology mainly falls within the class of regular languages (see Beesley and Karttunen 2003 and references therein); even challenging non-concatenative morphological phenomena, such as circumfixation and root-and-pattern morphology, can be characterized by regular means (Beesley and Karttunen, 2000, 2003). There is one exception to this generalization: productive reduplication can be characterized only by a more powerful context-sensitive grammar.[2] However, morphologists have so far

[2] Beesley and Karttunen (2000, p. 6) show, however, that, if one can assume that "there are only a finite number of words subject to reduplication (no free compounding)", even total reduplication, e.g. as in Malay, can be captured by regular means.

not discovered any other phenomenon besides reduplication which requires going beyond regular languages: words do not contain deeply nested dependencies or long-distance dependencies, for example. It seems, then, that natural language morphology falls within the regular languages, with one exception: reduplication.

Natural language syntax is formally very different: it is pervaded by dependencies that require at least context-free power to describe. Partee *et al.* (1993, pp. 480ff.) provide a proof that English is not a finite-state language, using nested dependencies and the Pumping Lemma. The question of whether natural language syntax falls outside the class of context-free languages has been more difficult to answer, and early arguments for the inadequacy of context-free grammars for natural language syntax were shown to be flawed by Pullum and Gazdar (1982). Shieber (1985) provided a definitive proof that natural language syntax falls in the class of mildly context-sensitive languages (Joshi *et al.*, 1991; Vijay-Shanker and Weir, 1994), on the basis of data from Swiss German cross-serial dependencies. In fact, however, cross-serial dependencies seem to be the only phenomenon requiring more than context-free power for natural language syntax, and it may be that languages without cross-serial dependencies can be satisfactorily characterized in context-free terms.

Whether natural language syntax is mainly context-free or fully context-sensitive, the substantial formal differences between morphology and syntax remain entirely unexplained if the full computational power of the syntactic system underlies morphology. If morphology has the full power of syntax, why are there no clear morphological equivalents of unbounded or nested dependencies? It is of course possible that morphology does have the full expressive power of syntax, but we simply have not yet come across languages with unbounded morphological dependencies. However, this would be surprising, given that unbounded dependencies are syntactically prevalent across the world's languages. Similarly, why do we fail to find reduplication in the syntax, if there is no important formal distinction between morphology and syntax? These questions do not arise if we maintain a theoretical distinction between words and phrases.

In this paper, we will show that it is not necessary to remove the word/phrase distinction or the lexicon/grammar distinction to capture constructional effects, although we agree that there are important generalizations involving constructions of all sizes that must be captured at both grammatical and semantic levels. In keeping with much other work in Lexical Functional Grammar (see particularly Bresnan, 2001, Chapter 6), we believe that the proper grammatical level for characterizing similarities across constructions is a level like LFG's *functional structure* (f-structure), which represents abstract syntactic relations such as subject, object, and adjunct, as well as syntactic features such as person, number, gender, case, tense, and aspect. In contrast, the *constituent structure* (c-structure) tree encodes word order, phrasal dominance, and grouping; it is the level at which the difference between words and phrases is represented, in keeping with the *Lexical Integrity Principle*. As Bresnan (2001, p. 93) observes, "… LFG's lexical integrity principle implies that while morphemic words and syntactic phrases are different types of forms of expression in c-structure, they may carry the same types of information in f-structure. In other words, these different forms of expression – words and phrases – may be functionally equivalent (in terms of f-structure content)."

An early statement of Lexical Integrity was provided by Simpson (1983, p. 74):

(4) *Lexical Integrity* (Simpson, 1983, p. 74)
No constituent structure rule may order any element into or out of lexical categories such as N, A, V. That is, constituent structure rules are blind to the internal structure of lexical categories.

Subsequent work within LFG has adopted Lexical Integrity as a fundamental principle differentiating word-internal structure from phrasal syntax, as in (5), and establishing words as indivisible, undecomposable units at c-structure as in (6):

(5) *Lexical Integrity* (Bresnan and Mchombo, 1995, p. 181)
Words are built out of different structural elements and by different principles of composition than syntactic phrases.

(6) *Lexical Integrity* (Bresnan, 2001, p. 93)
Morphologically complete words are leaves of the c[onstituent]-structure tree and each leaf corresponds to one and only one c[onstituent]-structure node.

In a series of papers, Booij (2005a,b, 2009) provides substantial evidence for the Lexical Integrity Principle from a Construction Grammar viewpoint. His observations are an excellent fit with the LFG view of Lexical Integrity, which assumes that syntactic rules have access to certain (f-structural) aspects of word-internal structure, but not to other (c-structural) aspects.

The c-structure/f-structure distinction is crucial to the LFG-theoretic understanding of the *Strong Lexicalist Hypothesis*, which is essentially what Lexical Integrity captures. The Strong Lexicalist Hypothesis states that syntactic rules of phrase formation cannot access any parts of words. This is a generalization of the *Weak Lexicalist Hypothesis*, which states that syntactic rules cannot access derivational morphology, but allows access to inflectional morphology. Marantz (1997) has argued that lexicalism is untenable based on the view that morphology can make complex syntactic contributions; this is the foundational doctrine of Distributed Morphology. However, although Marantz (1997) has often been taken as arguing against lexicalism *tout court*, his arguments actually depend on his particular conception of syntax, which does not distinguish constituent structure from functional structure.

Lexical Integrity as a principle of c-structure does not disallow words from making complex contributions at functional structure; this means that it is possible for individual, morphologically complex words to express the same information as multi-word expressions. For example, the future tense can be expressed with verbal morphology as in the French example in (7), or with a future auxiliary as in the English example in (8).

(7) Il arrivera.
he arrive.FUTURE
'He will arrive.'

(8) He will arrive.

In (7), the future tense is realized directly on the main verb. In contrast, the future is expressed with the morphologically independent auxiliary *will* in (8); see Ackerman and Webelhuth (1998) and Ackerman and Stump (2004) for more discussion of examples of this type.

To take another example, the Swedish singular indefinite marker is a determiner, as in English, realized as *en* or *ett* depending on the gender. However, the Swedish definite marker is a morphologically bound suffix on the noun. Example (9) shows the periphrastic indefinite + noun combination, while (10) shows the same noun with a definite suffix:

(9) En väg kan vara mycket lång.
a road can be very long
'A road can be very long.'

(10) Vägen hem var mycket lång.
road.the home was very long
'The road home was very long.'

The periphrastic expressions *will arrive, en väg* and *a/the road* are comparable to the synthetic *arrivera* and *vägen*. The periphrastic and synthetic forms alike contain information that is syntactically relevant.

The dual nature of syntactic structure in Lexical Functional Grammar (LFG: Bresnan 2001; Dalrymple 2001; Falk 2001) captures both the commonalities and the differences between words and phrases, as argued at length by Simpson (1983), Mohanan (1994, 1995), Bresnan and Mchombo (1995), Matsumoto (1996), Bresnan (2001), and many others. Constituent structure represents surface word order and phrasal grouping; in accordance with Lexical Integrity, morphologically bound information about tense and definiteness is 'invisible' at c-structure, in the sense that the information is not hosted by separate c-structure nodes. At functional structure, words and phrases can make similar or identical syntactic contributions, accounting for the similarities between words and phrases at this more abstract syntactic level: as Bresnan (1998) puts it, morphology competes with syntax in cases like (7)–(10), specifying similar grammatical structure by different morphological means. LFG's definition of Lexical Integrity entails that individual morphemes may contribute to functional structure, while the smallest unit visible at constituent structure is the word.

A comparison of English *the road* and Swedish *vägen* shows that morphology can contribute information directly to the f-structure without violating Lexical Integrity at c-structure. Lexical entries for *the*, *road*, and *vägen* are given in (11)–(13):

(11) the D $(\uparrow \textsc{definite}) = +$

(12) road N $(\uparrow \text{PRED}) = \text{‘road’}$
$(\uparrow \text{NUMBER}) = \text{SG}$
$(\uparrow \text{PERSON}) = 3$

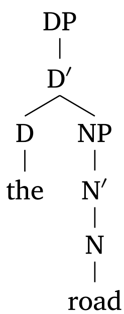
(13) vägen N $(\uparrow \text{PRED}) = \text{‘road’}$
$(\uparrow \text{NUMBER}) = \text{SG}$
$(\uparrow \text{PERSON}) = 3$
$(\uparrow \text{DEFINITE}) = +$
$(\uparrow \text{GENDER}) = \text{COMMON}$

C-structures and f-structures for *the road* and *vägen* are given in (14):

(14)

$$\begin{bmatrix} \text{PRED} & \text{‘road’} \\ \text{DEFINITE} & + \\ \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \\ \text{GENDER} & \text{COMMON} \end{bmatrix}$$

(15)



$$\begin{bmatrix} \text{PRED} & \text{‘road’} \\ \text{DEFINITE} & + \\ \text{NUMBER} & \text{SG} \\ \text{PERSON} & 3 \end{bmatrix}$$

Despite the very different c-structures for *the road* and *vägen,* the f-structures are almost identical, the only difference being that English nouns do not bear gender. In LFG, syntactically relevant information can be contributed to the f-structure by bound morphology, even though the internal complexity of words is invisible at c-structure. Lexical Integrity is therefore maintained at c-structure.

Multi-word constructions, handled straightforwardly in Construction Grammar, pose a direct problem for a framework like LFG that adopts the Strong Lexicalist Hypothesis. Most work referring to Lexical Integrity in LFG has focused on the internal structure of words. However, the principle is more general than that: Lexical Integrity dictates that there is a one-to-one mapping between fully inflected words and c-structure nodes. In other words, units smaller than a word cannot be inserted into c-structure and units bigger than words cannot be inserted into c-structure. Each word is independent and corresponds to a single c-structure node. This is problematic when a particular combination of words gives rise to a meaning non-compositionally (or perhaps semi-compositionally), as has been argued to be the case for expressions such as those in (1)–(3).

This paper proposes that the key to capturing constructional effects in LFG is the observation, outlined above, that words and phrases can make identical contributions to f-structure. Given the Glue Semantics approach to compositional semantics (Dalrymple, 1999, 2001; Asudeh, 2012), which allows terms for semantic composition to be specified based on f-structures, this equally means that words and phrases can make identical semantic contributions. We provide a means of expressing commonalities in functional structure and semantics across linguistic units of various sizes through the means of LFG's *templates* (Dalrymple *et al.*, 2004; Asudeh, 2012), bundles of grammatical descriptions, which can be associated with parts of words, with words, or with phrases. Templates can be defined in terms of other templates, thus allowing us to express similarities and differences between constructions, whether they are expressed by a single word or a phrase. In this way, we account for the similarities between words and phrases which have been a focus of work within Construction Grammar, but within a framework which also incorporates the differences between words/phrases and lexicon/grammar as a fundamental architectural principle. We believe that our proposals are valid no matter what theory of morphology is adopted, and no matter whether the term "lexicon" refers to a list of words, a list of morphemes, or a list of all unpredictable form-meaning pairs whatever their size.

The paper is structured as follows. In Section 2, we present three similar constructions in English, Swedish, and Dutch and show that the constructions are expressed differently in the different languages. In Section 3, we present templates and show how they can be used to express generalizations. In Section 4, we present our formal analysis in terms of LFG with Glue Semantics. In Section 5, we briefly consider a further generalization of the theory in terms of linking theory, which concerns the instantiation of grammatical functions based on argument structure. Section 6 is the conclusion. The paper ends with a set of formal appendices, which includes complete Glue proofs for three examples.

2 CONSTRUCTIONS EXPRESSED IN WORDS AND PHRASES

We first present our view of constructions in LFG. As an illustration of our view, we examine variants of the traversal construction in English, Swedish, and Dutch, showing that different properties of the construction – the phrasal configuration, some combination of words in the construction, or both – are responsible for its meaning. Section 2.1 discusses the English *way*-construction, which is signalled by the presence of the word *way*. Section 2.2 discusses the Swedish counterpart of this construction, the Directed Motion Construction, which is signalled not by a particular word, but by a special phrasal configuration. Finally, Section 2.3 presents two Dutch constructions, the *weg*-construction and the Transition to Location Construction; the *weg*-construction is similar to its English counterpart, in that the construction is signalled by the presence of the word *weg*. The Dutch Transition to Location Construction is different from both English and Swedish, in that there is no special word or phrasal configuration to signal the construction. Each of these patterns involves specification of the constraints associated with the construction in a different way: associated with a word other than the head predicate (English *way* or Dutch *weg*); associated with a special phrase structure rule (the Swedish Directed Motion Construction); or associated with the main, head predicate of the construction (the Dutch Transition to Location Construction).

2.1 *Signalled by a non-head word*

It has been argued that the English *way*-construction in (16) deserves a constructional analysis rather than a compositional one, since the construction implies directed motion even though none of the individual words in *way*-examples necessarily denote motion (Jackendoff, 1992, 1990; Goldberg, 1995).[3] Our analysis associates constraints on the form and meaning of the construction with the word *way*.

The analysis must capture several generalizations about this construction. The action denoted by the verb *elbow* does not normally involve traversal, though in example (16) this meaning is present.

(16) Sarah elbowed her way through the crowd.

In fact, for most English speakers the English *way*-construction has two closely related meanings, one involving means and one involving manner (Jackendoff 1990, p. 215, Goldberg 1995, pp. 202–212), though Goldberg (1995, pp. 202–203) points out that the manner interpretation is not available for all speakers. Examples (17) and (18) both involve an event denoted by the main verb (whistling or elbowing) and its relation to a second event of traversal of a path. The verb *elbowed* in example (17) specifies the means by which Sarah managed to traverse the crowd: the traversal was made possible by the elbowing action. For those who allow the manner interpretation, the verb *whistled* in example (18) specifies the manner in which the traversal of the room took place: Sarah whistled while crossing the room.

(17) Means: Sarah elbowed her way through the crowd. (traversed the crowd by means of elbowing)

(18) Manner: Sarah whistled her way across the room. (traversed the room while whistling)

Our analysis allows us to specify what these meanings have in common and how they differ; it also allows the statement of cross-linguistic similarities and differences in similar constructions in other languages.

Jackendoff (1990, p. 216) and others have claimed that the possessor in the English *way*-construction must be coreferential with the subject, and indeed, in an overwhelming number of cases, this generalization holds. However, we have found examples which counterexemplify this claim:[4]

---

[3] Marantz (1992) informally sketches an alternative view of the *way*-construction, which attempts to derive properties of the construction from facts about direct complementation and secondary predication.

[4] A reviewer suggests that the possessor must nevertheless somehow be ‘linked’ to the subject, even if coreference is too strong a constraint. If so, the link need not be narrowly syntactic, as already witnessed by the total lack of a

(19) He had bought his son's way into an exclusive military academy normally reserved for the gentry and had outfitted him in style.
`www.samizdat.com/hero7.html` (retrieved May 27, 2013)

(20) As ambassador, Chesterfield negotiated Britain's way into the Treaty of Vienna in 1731.
`www.aim25.ac.uk/cgi-bin/frames/fulldesc?coll_id=2117{&}\`
`inst_id=86`
(retrieved May 27, 2013)

Furthermore, the noun *way* in the *way*-construction can be modified (Jackendoff 1990, p. 217, Goldberg 1995, p. 206):

(21) In these last twenty years Richard Strauss has flamed his meteoric way into our ken – and out of it. (Buchanan, 1918)

An analysis of the construction must be able to derive a meaning for these examples as well; our analysis does.

2.2 *Signalled by a special phrasal configuration*

Toivonen (2002) discusses the Swedish Directed Motion Construction (DMC). The DMC, which is exemplified in (22), consists of a subject, a verb, a weak reflexive (coindexed with the subject), and a directional PP.

(22) Sarah armbågade sig genom mängden.
S. elbowed SELF through crowd.DEF
∼'Sarah elbowed her way through the crowd.'

The Swedish DMC is very similar in meaning and use to the English *way*-construction, but the DMC does not include any word such as

---

pronoun in (20) and as further emphasized by the fact that the *way*-phrase may simply be an indefinite:

(i) Daimler bought a way into the Chinese market and then removed Chrysler from it.
`www.thetruthaboutcars.com/2013/04/`
`jeep-eyeing-chinese-cherokee-production/` (retrieved May 28, 2013)

Therefore, the link between the subject and the possessor is at best something akin to 'bridging' (Haviland and Clark, 1974), although the latter term is typically associated with definites, not indefinites. Bridging is normally viewed as a pragmatic inference. We return to this issue in footnote 16 below, after the formal analysis has been introduced.

way to flag the construction. Instead, the construction is distinguished by the strict requirement for the presence of certain constituents, restrictions on the individual constituents, and perhaps most interestingly, by a word order quirk at odds with the rest of Swedish grammar (Toivonen, 2002). This word order quirk is only seen in DMC expressions that contain a verbal particle. Consider (23a–b):

(23) a. Jonas knuffade sig in i mängden.
J. pushed SELF in inside crowd.DEF
'Jonas pushed his way into the crowd.'

b. Jonas knuffade in dig i mängden.
J. pushed in you inside crowd.DEF
'Jonas pushed you into the crowd.'

Verbal particles in Swedish (such as *in*) are normally adjoined to the verb, and must precede the direct object, as in (23b) (Toivonen, 2003). However, in the DMC, the particle may not adjoin to the verb; instead, it is a part of the PP, *in i mängden*, and follows the reflexive, for example *sig* in (23a).

Toivonen (2002) considers two distinct analyses of the DMC. One proposed analysis is constructional, in the sense that the DMC meaning is associated directly with a syntactic frame; we adopt an analysis of this sort, as we discuss below. The other analysis is purely lexical: the DMC verb is related to another verb via a lexical redundancy rule. The DMC verb carries very detailed specifications about what types of arguments it must take. Two facts disfavour the second analysis, which ties the DMC to the verb: first, DMC verbs cannot participate in any kind of derivational morphology. They cannot be turned into nouns or adjectives, for example. The DMC verbs can also not be passivized. Second, the most striking distinguishing feature of the DMC is the peculiar word order constraint mentioned above. Prepositional particles cannot appear in the normal, pre-object particle position in the DMC. This is what distinguishes the DMC from resultatives, for example. For these reasons, we prefer an analysis which connects the DMC meaning with a specific phrase structural configuration.

2.3 *Signalled by the head*

Van Egmond (2006, 2009) shows that Dutch has two constructions that indicate traversal of a path. One construction contains the word *weg* 'way' (24), and the other does not (25).

(24) Wij worstelen ons een weg door de menigte.
we wrestle ourselves a way through the crowd
'We are wrestling our way through the crowd.'

(25) Janneke bluft zich uit de benarde situatie.
J. bluffs SELF out the awkward situation
∼'Janneke bluffs her way out of the awkward situation.'

The *weg*-construction exemplified in (24) is also discussed in Verhagen (2003).

Although the two Dutch constructions are similar in meaning, van Egmond (2006, 2009) shows that they nevertheless have distinct interpretations. She calls the type with *weg* (24) the '*weg*-construction', and the type without *weg* (25) the 'Transition to Location Construction' (TLC). The *weg*-construction describes an incremental traversal of a path by means of (or while) performing the activity denoted by the verb. The traversal and the activity denoted by the verb are coidentified: the construction describes a simple event. The TLC, on the other hand, describes a transition to a stative location by means of performing the activity denoted by the verb, without necessarily traversing a path. The traversal and the activity denoted by the verb are two subevents that are not necessarily coextensive. For example, in (25), the bluffing event can take place at a point in time preceding the event in which the subject gets out of the awkward situation.

We are here interested in the TLC, as it provides an interesting contrast to the English and Swedish constructions introduced above. Unlike the English *way*-construction, the TLC does not contain a specific word (such as *way*) that 'flags' the construction, and unlike the Swedish DMC, the Dutch TLC does not display special syntax: the word order follows the rules of regular Dutch syntax.[5] We propose that the TLC information is associated with the verb. No matter which verb is included in the construction, the TLC requires exactly three arguments: a subject, a reflexive direct object and a postpositional oblique.

[5] The PP in the Dutch examples is a prepositional phrase, not a postpositional phrase. Directional PPs normally contain postpositions in Dutch, while stative locations are prepositional PPs. The fact that the TLC contains postpositions and not prepositions follows from van Egmond's (2006; 2009, pp. 99–101) analysis of the TLC as an expression of transition to a stative location.

The TLC changes the basic argument requirements of the verb, and is in this way comparable to passives, causatives and applicatives, all of which also have relation-changing characteristics. Relation-changing processes are standardly treated in LFG as alternative ways of mapping thematic roles to syntactic roles.

2.4 *Summary*

These expressions from Swedish, English and Dutch have in common a core part of their meanings and also the fact that their meanings are not straightforwardly predictable from the meanings that their parts have in other contexts. Each of them has been noted to be problematic for Lexical Integrity. Below, we present analyses of these constructions that capture the relevant data while preserving Lexical Integrity. The Swedish DMC, the English *way*-construction, and the Dutch *weg*-construction and TLC have distinct syntactic realizations. However, as has been argued in detail by van Egmond (2006, 2009), Toivonen (2002) and Verhagen (2003), there are nevertheless strong reasons for treating them as distinct realizations of the same 'construction'. The basic similarity lies in their meaning, as is evident from the fact that the expressions translate into each other across the languages. The expressions entail *traversal*, even though this sense is not necessarily contributed by the verb. The verb does not need to be a motion verb. Our task is to capture the similarities between the constructions, while at the same time modelling their differences. The constructions crucially differ in which formal element carries the traversal meaning. We assume that the traversal meaning is signalled by the word *way* in English, as this word is necessarily present. The Swedish DMC does not contain a specific word signalling the construction; all words in the Swedish DMC are exchangeable, since even the reflexive changes to agree with its antecedent. Instead, the construction is flagged by its word order. We therefore assume that the information that is specific for the traversal reading is tied to a phrase structure rule. Finally, the Dutch TLC is not associated with a specific word or peculiar word order. We therefore make the assumption that the clausal head, the verb, is the locus of the relevant information.

3 ENCAPSULATING GENERALIZATIONS THROUGH TEMPLATES

Our proposal allows for cross-linguistic generalizations to be captured by the use of *templates* to encode complex syntactic descriptions and the relations among them. Templates can be associated with objects of various sizes, from parts of words to phrases. This does not violate LFG's Lexical Integrity Principle, as words are still intact and independent at c-structure.

3.1 *Background*

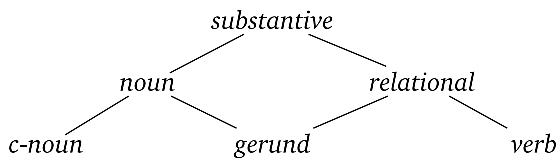
An LFG template is nothing more than a named functional description, where the latter is a set of equations that describe linguistic structures. For any LFG grammar defined in terms of templates, we could construct a completely equivalent grammar which does not use templates, simply by replacing each template with the description that it abbreviates: by doing this, the same grammatical descriptions would be associated with words and phrases in each of the two grammars, and the grammars would produce the same c-structures and f-structures for the words and phrases of the language. Importantly, however, the grammar without templates would lack the means of expressing generalizations across lexical entries and grammar rules which templates make available.

Functional descriptions most notably describe f-structures, but can in principle contain descriptions of any linguistic structure in LFG's Correspondence Architecture (Kaplan, 1987; Asudeh, 2006). A template associates a name with a given functional description, such that the description can be invoked throughout the lexicon, as originally envisioned (Dalrymple *et al.*, 2004), or, more generally, anywhere in the grammar, as we propose. Templates thus not only allow generalizations to be captured within the lexicon, but also across the lexicon and the rest of the grammar. It is in this sense that templates can be construed as an important component of the lexicon–grammar interface.

Template definitions may contain reference to other templates. This effectively creates a hierarchy of templates, similar to the perhaps more familiar type hierarchies of Head-Driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1987, 1994; Ginzburg and Sag, 2000) and Sign-Based Construction Grammar (SBCG; Michaelis, 2010; Sag, 2010; Boas and Sag, 2012). However, there are some noteworthy differences between templates and types. First, type hierarchies represent relations between structures, whereas template hierarchies represent relations between descriptions of structures. This means that templates do not appear in the actual structures of the theory, but only in descriptions that the structures must satisfy. This contrasts with types in HPSG, where each structure (a directed acyclic graph) is associated with a type.

Second, type hierarchies in HPSG and SBCG represent inheritance in an *and/or* semilattice. The daughters of a type represent disjoint subtypes (*or*). Multiple mothers for a type represent conjoined supertypes (*and*). For example, consider the following type hierarchy from Malouf (1998):

(26)



The type *substantive* is a subtype of the root type *head* (not shown here). Its two daughters, *noun* and *relational*, disjointly partition the supertype. Each of the types *noun* and *relational* in turn has two daughters that disjointly partition the type. However, the type *gerund* is common to both and constitutes a conjunction of the supertypes: a *gerund* object is both a *noun* object and a *relational* object.

Template hierarchies do not represent inheritance, but rather inclusion. If a template A dominates a template B, then the description that A labels appears in the description that B labels. The semantics of template invocation, denoted by the prefix @ in a description, is just substitution. For example, given the 3SG template in (27) below, the lexical entries in (28a) and (28b) are strictly equivalent.

(27) 3SG := ($\uparrow$ SUBJ PERS) = 3
($\uparrow$ SUBJ NUM) = SG

(28) a. *laughs* ($\uparrow$ PRED) = 'laugh⟨SUBJ⟩'
($\uparrow$ SUBJ PERS) = 3
($\uparrow$ SUBJ NUM) = SG

b. *laughs* ($\uparrow$ PRED) = 'laugh⟨SUBJ⟩'
@3SG

It is clear from this example that a template is nothing more than an abbreviation for an LFG description. Throughout this paper, we use the term 'hierarchy' in reference to templates to mean 'inclusion hierarchy', not 'inheritance hierarchy'.

Descriptions in LFG support the boolean operations of conjunction, disjunction and negation. Templates therefore also support these operations. For example, the 3SG template can be negated in a lexical entry:

(29) *laugh* (↑ PRED) = 'laugh⟨SUBJ⟩'
¬@3SG

The lexical entries for *laughs* and *laugh* would thus both be daughters of the template 3SG in a template hierarchy, because both entries include the template, even if one negates it and the other does not:

(30)
3SG
laugh *laughs*

This emphasizes the difference between a hierarchy that represents inheritance, as in HPSG or SBCG type hierarchies, versus a network that represents inclusion, as in LFG template hierarchies. It would not make sense for both *laugh* and *laughs* to *inherit* from a 3SG type object, but both words can nevertheless *include* the description (with or without negation) that is labelled by the 3SG template.

Templates can also be parametrized, where the parameters are stated as arguments to the template. For example, the template in (31) could be used in the lexical entry for any intransitive verb, such that the entry for *laughs* could be rewritten as in (32).

(31) INTRANS(P) := (↑ PRED) = 'P⟨SUBJ⟩'

(32) *laughs* @INTRANS(laugh)
@3SG

The lexical entry for *laughs* in (32) is still strictly equivalent to the one in (28b), but the templates bring to the fore the generalization that the only idiosyncratic information is what is contributed by the verb root. For example, the entry for *yawns* would differ only in the argument to the parametrized INTRANS template ('yawn' instead of 'laugh').

The question potentially arises of where templatic information is stored in an LFG grammar.[6] Since templates are abbreviations for

[6] We thank one of our anonymous reviewers for raising this point.

functional descriptions, they need to be accessible to any component of the grammar that makes use of f-descriptions, in particular the lexicon and c-structure rules. In the Xerox Linguistic Environment (Crouch *et al.*, 2012), the standard grammar development platform for LFG, each grammar can have a section that defines templates; this component is loaded with the lexicon and rule components.

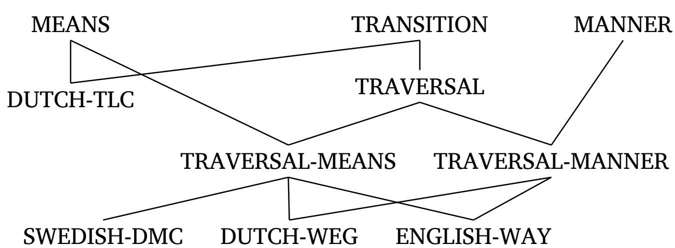
Lastly, as we will see presently, template hierarchies need not have a single root: there is no need for a most general template whose description is included in all of the other templates in the network. Templates can include references to other templates, or they can stand alone, as we will see in our definition of the Transition Template Hierarchy.

3.2 *The Transition Template Hierarchy*

We propose a single theory of constructions that uses existing LFG mechanisms to capture commonalities involving the traversal/result construction in English, Swedish, and Dutch. Our theory preserves the intuition that similar constructional specifications can be associated with different components of the construction. The English *way*-construction is driven by lexical specifications for *way*, together with general phrase structural facts about English. Similarly, the Dutch *weg*-construction is driven by lexical specifications for *weg*. The Swedish DMC is driven by a specific phrase-structural configuration. The Dutch TLC is associated with specifications on the verbal predicate.

The template hierarchy we assume is represented in (33):[7]

(33)



The template TRANSITION contains material that is common to the Swedish, Dutch, and English constructions; it encodes information

[7] Note that this hierarchy is not directly represented as a hierarchy in a grammar, but rather is a representation of the inclusion relations between the relevant templates.

about agency and causation. MEANS and MANNER are two other general templates that specify information about the means or manner of the event. The TRAVERSAL template is defined in terms of the more general TRANSITION template, as represented by the line connecting them, which means that TRAVERSAL includes all of the information associated with the TRANSITION template while also contributing some information specific to TRAVERSAL. TRAVERSAL in turn appears as a part of the definition of both the TRAVERSAL-MEANS template and the TRAVERSAL-MANNER template.

The templates TRAVERSAL-MEANS and TRAVERSAL-MANNER provide different ways of adding information to the TRAVERSAL template, supplying the information that the main verb denotes either the means or the manner in which the path traversal is achieved. The Swedish DMC has the means interpretation (Toivonen, 2002, p. 318), and so we treat it as associated with the TRAVERSAL-MEANS template; the manner interpretation may be available dialectally, but we do not treat this variation here. The templates SWEDISH-DMC, DUTCH-WEG, and ENGLISH-WAY contribute additional language-specific information to these templates, as we will see. Finally, the DUTCH-TLC is another language-specific template which draws together information from MEANS and TRANSITION.

In keeping with LFG's focus on typological generalizations, this templatic approach sheds light on differences and similarities in constructional effects across languages. Information that is shared across constructions in a language can be stated in non-terminal nodes of the template hierarchy and is inherited by specific constructional templates. Similarly, the fact that grammatical information may be shared by constructions in different languages is captured by calls in language-specific constructional templates to the same more general templates in the hierarchy. This approach to grammatical variation is of long standing in the HPSG and now Sign-Based Construction Grammar traditions; see Pollard and Sag (1994, pp. 57–59) and discussion in Sag *et al.* (2012).

For example, the English *way*-construction (on one interpretation), the Swedish Directed Motion Construction, and the Dutch *weg*-construction all share a meaning component that concerns means of traversal of a path and therefore all call the TRAVERSAL-MEANS template in our hierarchy. The template hierarchy thus illustrates a typological space of possibilities. We leave it an open question whether these possibilities are made available to the language learner in Universal Grammar or whether the hierarchy is simply a convenient way to describe and classify the relevant constructions that can be observed cross-linguistically. For clarity of exposition, we have chosen to label the terminal templates in the hierarchy with language-specific names, such as 'SWEDISH-DMC', but the constellation of template calls and additional information that is realized in this template may of course also be instantiated in this exact form in other languages. Therefore, more accurate but less user-friendly names for the terminal templates would be names like 'TRAVERSAL-MEANS-PHRASAL', 'TRAVERSAL-MEANS-NON-HEAD-MARKED', 'TRAVERSAL-MEANS-HEAD-MARKED', and so forth.

4 FORMAL ANALYSIS

4.1 *Phrase structurally flagged constructions*

4.1.1 The phrase structure rule

Turning first to the Swedish DMC, we propose that this construction is most elegantly analyzed with the following construction-specific phrase structure rule, which makes crucial use of a call to the template SWEDISH-DMC:[8]

(34)
$$\begin{array}{llccc} \mathrm{V}' & \longrightarrow & (\mathrm{V}^0) & \mathrm{NP} & \mathrm{PP} \\ & & \uparrow = \downarrow & (\uparrow \textsc{obj}) = \downarrow & (\uparrow \textsc{obl}) = \downarrow \\ & & & (\downarrow \textsc{prontype}) = & \\ & & & \textsc{simplex-reflexive} & \\ & & & @\textsc{swedish-dmc}(\uparrow \textsc{pred fn}) & \end{array}$$

The template call appears on the NP node. This is a partially arbitrary

[8] In some of the literature on templates (Dalrymple *et al.*, 2004), distinct arguments to parametrized templates are separated by spaces, but we follow the convention of Asudeh (2012) of separating arguments to templates explicitly using commas. Spaces, in our notation, do not indicate distinct arguments, but rather serve their standard role in LFG feature specifications. Thus, in this c-structure rule, the template takes a single argument (not three): the f-structure described by ↑ PRED FN.

decision; the call could instead appear on another node. We chose the NP, as we see the reflexive as a signal of the construction. Note, however, that the reflexive changes according to the person and number of the subject: it is not a fixed lexical marker of the construction, which makes it different from *way/weg*. By convention, template calls are marked by the at sign '@'. The SWEDISH-DMC template takes a single argument, the value of the PRED FN of the V′; we provide more information about this template in Section 4.1.2.

Notice that FN is not itself a semantic form, but rather part of a semantic form; the attribute FN and argument designators such as ARG1 allow reference to the components of a semantic form (Crouch *et al.*, 2012) according to the following pattern:[9]

(35) [PRED 'FN⟨ARG1,ARG2,...⟩NONARG1,NONARG2,...']

The specifications in (36a,b) are thus equivalent:

(36) a. (f PRED) = 'elbow⟨(↑ SUBJ),(↑ OBJ)⟩'

b. (f PRED FN) = elbow
(f PRED ARG1) = (↑ SUBJ)
(f PRED ARG2) = (↑ OBJ)

Use of the attribute FN thus allows reference to the predicate name in PRED features, setting subcategorization aside. The implications of this are further discussed in Section 4.2 below.

We observe four important properties of our treatment of the SWEDISH-DMC. First, associating the template for this construction with a special phrase structure rule reflects the fact that only this particular configuration has the special meaning associated with the DMC.

Second, the NP and PP daughters of V′ in (34) are obligatory. Our theory assumes that optionality must be explicitly marked in phrase structure rules, as in computational LFG treatments (e.g. Crouch *et al.* 2012) and in contrast to theoretical positions that allow generalized optionality (e.g. Bresnan 2001). The V^0 node is optional, since the verb need not appear there: the Swedish finite verb appears in I rather than V.

[9] The arguments of the semantic form are separated into thematic and non-thematic arguments, indicated by ARG_n and NONARG_n respectively.

Third, we must explicitly state the fact that the NP is a simplex reflexive, such as *sig*, and not just any kind of NP or even a complex reflexive (e.g. *sig själv*).[10]

Fourth, the construction requires an OBL(IQUE) phrase. Since the OBL must be realized as a post-object PP, it cannot also be realized as a pre-object particle. Post-object particles are projecting, intransitive prepositions (Jackendoff, 1973; Toivonen, 2003). The OBL must be directional, which we capture by referring to a PATH feature in semantic structure, as in (42) below.

4.1.2 The SWEDISH-DMC template

Semantically, the Swedish DMC and the English *way*-construction involve an event characterized by the main verb in the construction and a second event involving traversal of a path. The basic template TRANSITION is defined as follows:

(37) TRANSITION :=

$$\lambda R \lambda x \lambda e \lambda e'.R(e) \ \wedge \ agent(e) = x \ \wedge \ cause(e') = x :$$
$$(\uparrow_\sigma \text{REL}) \multimap (\uparrow \text{SUBJ})_\sigma \multimap (\uparrow_\sigma \text{EVENT1}) \multimap (\uparrow_\sigma \text{EVENT2}) \multimap \uparrow_\sigma$$

Templates encoding syntactic information and expressing syntactic generalizations are defined as sets of functional equations, as described by Dalrymple *et al.* (2004). However, since our concern is the syntax–semantics interface and meaning differences among constructions, we define this template with a *meaning constructor* (Dalrymple, 1999, 2001; Asudeh, 2004, 2012), which provides part of the common meaning for the traversal/result construction in English, Swedish, and Dutch.[11] This meaning constructor requires:

- a REL meaning R specifying the nature of the event e, which is provided by the verb in the construction; for *Bill elbowed his way through the crowd*, e is required to be an event of elbowing, and so R is the predicate *elbow*;
- a meaning x for the subject of the main verb, which is interpreted as the agent of e and the causer of the transition event e';

[10] This information could be moved into the SWEDISH-DMC template itself. We leave it on the phrase structure rule simply to highlight it.

[11] The subscripted σ's in meaning constructors indicate mappings to semantic structure, a level in LFG's grammatical architecture (Dalrymple, 2001; Asudeh, 2012).

- two event variables e and e', associated with the semantic attributes, EVENT1 and EVENT2, representing the event denoted by the verb and the transition event.

This basic meaning is augmented by other meaning constructors in the template hierarchy. Our characterization of the subject of the main event as an agent of the event e and a causer of the transition event e' follows Goldberg (1995, pp. 212–213), who claims that the motion in the way-construction must be self-propelled. However, Jackendoff (1990, p. 216) suggests that although the means interpretation is necessarily tied to deliberate action, the manner interpretation is also compatible with action that is not deliberately performed. Examples such as (38), which has a manner and not a means interpretation, are better characterized by Jackendoff; in this example, e is an event of bleeding, which is not associated with an agent:

(38) Baxter's wife said her son bled his way into the ambulance painlessly.
`newvoices.org/2005/03/08/0089/` (retrieved May 28, 2013)

To account for these examples, it may be better to refer to the highest thematic argument of the main event e rather than explicitly referring to the agent. We leave further exploration of this issue for future research, and provisionally encode the relevant argument of the main event as an agent.

The template hierarchy in (33) encodes the fact that the template TRAVERSAL calls the template TRANSITION, with the effect that TRAVERSAL incorporates all of the information in TRANSITION as well as specifying some additional information. The TRAVERSAL template is defined in (39):

(39) TRAVERSAL :=
@TRANSITION
$\lambda P \lambda e'.P(e') \wedge \mathit{traversal}(e')$:
$[(\uparrow_\sigma \text{EVENT2}) \multimap \uparrow_\sigma] \multimap [(\uparrow_\sigma \text{EVENT2}) \multimap \uparrow_\sigma]$

The first line in the definition of TRAVERSAL contains the call to the template TRANSITION, marked as in (34) with the at sign '@'. The second line adds the information that e' is a traversal event. In technical terms, this meaning constructor behaves as a modifier on the predication associated with the transition event.

In turn, the TRAVERSAL-MEANS template is defined simply by calls to the TRAVERSAL template and the MEANS template:

(40) TRAVERSAL-MEANS := @TRAVERSAL
@MEANS

The MEANS template is given in (41):

(41) MEANS :=
$\lambda P \lambda e \lambda e'.P(e)(e') \wedge means(e') = e :$
$[(\uparrow_\sigma \text{ EVENT1}) \multimap (\uparrow_\sigma \text{ EVENT2}) \multimap \uparrow_\sigma] \multimap$
$[(\uparrow_\sigma \text{ EVENT1}) \multimap (\uparrow_\sigma \text{ EVENT2}) \multimap \uparrow_\sigma]$

The MEANS meaning constructor specifies that the event e represents the means of achieving the event e'. With respect to TRAVERSAL-MEANS, the main verb's event e is the means of achieving the event e' of traversing the path, as in an English example like *Sarah elbowed her way through the crowd* or the Swedish equivalent, where the traversal through the crowd is achieved by elbowing.

The SWEDISH-DMC template, specific to the Swedish Directed Motion Construction, is defined by reference to the template TRAVERSAL-MEANS. It also calls the syntactic subcategorization template TRANSITIVE-OBLIQUE, to be described in Section 4.2, and provides some additional material specific to the Swedish construction:

(42) SWEDISH-DMC(PFN) :=
@TRANSITIVE-OBLIQUE(PFN)
@TRAVERSAL-MEANS
$\lambda Q \lambda P \lambda y. \exists e. \exists e'. \exists z. P(e)(e') \wedge$
$theme(e') = y \wedge path(e') = z \wedge Q(z) :$
$[((\uparrow \text{ OBL})_\sigma \text{ PATH}) \multimap (\uparrow \text{ OBL})_\sigma] \multimap$
$[(\uparrow_\sigma \text{ EVENT1}) \multimap (\uparrow_\sigma \text{ EVENT2}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow \text{ OBJ})_\sigma \multimap \uparrow_\sigma$

The argument of the SWEDISH-DMC template is called 'PFN' in this definition;[12] it is passed as an argument to the TRANSITIVE-OBLIQUE template, which is defined in (49). Besides the two template calls, SWEDISH-DMC also contributes a meaning constructor to complete the meaning of the Swedish construction, which requires the following:

[12] The template argument PFN is meant to be mnemonic for PRED FN, since this will ultimately play the role of FN in the value of a PRED feature.

- a meaning Q depending on the OBL phrase, specifying the nature of the path traversed; for (22) (∼'Sarah elbowed her way through the crowd'), the path is required to go through the crowd;
- a meaning P, contributed by the main verb, specifying the nature of the event e denoted by the main verb and its relation to the transition event e'; for (22) (∼'Sarah elbowed her way through the crowd'), e is an elbowing event and is the means enabling the traversal event e';
- a meaning y for the object of the main verb, which is (syntactically) required to be a reflexive and hence to corefer with the subject of the main verb; y is the theme of e', the traversal event.

Our analysis produces the meaning in (43) for *Sarah armbågade sig genom mängden* 'Sarah elbowed SELF through the crowd'.

(43) $\exists e.\exists e'.\exists z.elbow(e) \wedge agent(e) = sarah \wedge cause(e') = sarah \wedge means(e') = e \wedge traversal(e') \wedge theme(e') = sarah \wedge path(e') = z \wedge through(z, \iota x.[crowd(x)])$

A full proof of the derivation of this meaning is given in the Appendix.

4.2 *Verb lexicon and basic subcategorization templates*

Our approach entails a potentially deep consequence for the theory of argument linking and subcategorization, because verbs in our approach specify default subcategorization through template calls in such a way that the subcategorization can be constructionally overridden. Thus, subcategorization is moved to the template component.

We have seen that the SWEDISH-DMC template provides a PRED specification with subcategorization frame and semantic specifications for the construction. This in turn means that the lexical entry for a verb must supply a default PRED and semantics which can be overridden when the verb is used in a construction like the *way*-construction.[13]

[13] Our analysis of the Swedish DMC and the English *way*-construction involves *replacing* rather than *modifying* the default semantic form of the main verb with the specifications provided by the construction. In the analysis of other constructions, it may be preferable to modify the semantic form via restriction (Kaplan and Wedekind, 1993; Asudeh, 2012) or other operators, as proposed for the analysis of complex predicates by Butt *et al.* (2003) (see also Butt and King 2005 on causatives).

We assume that the verb *elbowed/armbågade*, which appears in (16) and (22), is specified as follows:

(44) elbowed/armbågade V

$$\lambda e.elbow(e) : (\uparrow_\sigma \text{REL})$$

$$\left(\begin{array}{l} @\text{TRANSITIVE(elbow)} \\ \lambda R \lambda x \lambda y \exists e.R(e) \ \wedge \ agent(e) = x \ \wedge \ theme(e) = y : \\ (\uparrow_\sigma \text{REL}) \multimap (\uparrow \text{SUBJ})_\sigma \multimap (\uparrow \text{OBJ})_\sigma \multimap \uparrow_\sigma \end{array} \right)$$

The first line of the entry specifies the verb's semantic REL(ATION) in semantic structure, which it contributes on each occasion of its use. The second part of the entry specifies a default semantic contribution and subcategorization information, encoded by the template TRANSITIVE and the meaning constructor in the third line. This material effectively serves as a default, because unless some other part of the system specifies an alternative, constructional GF template, there is no way to check Completeness and Coherence and the structure fails.

The TRANSITIVE template takes a single argument, here 'elbow'. The definition of TRANSITIVE is stated with respect to an arbitrary argument PFN:[14]

(45) TRANSITIVE(PFN) := $(\uparrow \text{PRED}) = \text{`PFN}\langle(\uparrow \text{SUBJ}),(\uparrow \text{OBJ})\rangle\text{'}$

The argument PFN of the TRANSITIVE template appears in parentheses after the template name, and also appears in the definition of the template as the FN of the semantic form that is the value of the PRED feature. For the verb *elbow*, the call to the TRANSITIVE template passes in the argument 'elbow'. The template call @TRANSITIVE(elbow) is equivalent to the following equation:

(46) $(\uparrow \text{PRED}) = \text{`elbow}\langle(\uparrow \text{SUBJ}),(\uparrow \text{OBJ})\rangle\text{'}$

[14] For ease of explication, (45) specifies an active subcategorization frame for the verb, simplifying away from mapping theory issues and the possibility for passivization of this verb. We return to a discussion of the interaction of mapping theory and our theory of constructions in Section 5 below, where we propose a revised TRANSITIVE template which refers to argument structure roles rather than grammatical functions and which interacts appropriately with mapping theory.

We now turn to the default meaning constructor for *elbowed* given in (44), repeated here:

(47) $\lambda R \lambda x \lambda y \exists e.R(e) \wedge agent(e) = x \wedge theme(e) = y$:
$(\uparrow_\sigma \text{REL}) \multimap (\uparrow \text{SUBJ})_\sigma \multimap (\uparrow \text{OBJ})_\sigma \multimap \uparrow_\sigma$

This meaning constructor requires a REL R which is supplied by the verb (the REL for the verb *elbowed* is specified above as $\lambda e.elbow(e)$), a meaning x for the SUBJ, and a meaning y for the OBJ. For a sentence like *Bill elbowed Fred*, the meaning that is produced is, as desired:

(48) $\exists e.elbow(e) \wedge agent(e) = Bill \wedge theme(e) = Fred$

When the verb *elbowed* is used in the traversal construction, these default specifications are overridden by the specifications imposed by the construction, and the special constructional specifications are used instead.

Three further subcategorization templates and one control template are used in the analysis below:

(49) TRANSITIVE-OBLIQUE(PFN) :=
(↑ PRED) = ‘PFN⟨(↑ SUBJ),(↑ OBJ),(↑ OBL)⟩’

(50) TRANSITIVE-PREDICATIVE(PFN) :=
(↑ PRED) = ‘PFN⟨(↑ SUBJ),(↑ OBJ),(↑ XCOMP)⟩’

(51) DITRANSITIVE-OBLIQUE(PFN) :=
(↑ PRED) = ‘PFN⟨(↑ SUBJ),(↑ OBJ),(↑ OBJ_θ),(↑ OBL)⟩’

(52) OBJ-CONTROL := (↑ OBJ) = (↑ XCOMP SUBJ)

The TRANSITIVE-OBLIQUE template is called by the SWEDISH-DMC and ENGLISH-WAY templates, the TRANSITIVE-PREDICATIVE and OBJ-CONTROL templates are called by the DUTCH-TLC template, and the DITRANSITIVE-OBLIQUE template is called by the DUTCH-WEG template.

4.3 *Lexically flagged constructions*

The English *way*-construction relies on many of the same templates as the Swedish DMC. It is different in that it is completely regular in terms of phrasal structure, so no exceptional phrase structure rule is required. Rather, we assume the standard V′ rule for English, which already permits an NP OBJECT and a PP OBLIQUE. Evidence that the PP is an argument of the main verb and not a modifier of *way* comes from adverb placement: it is possible for an adverb to intervene between *way* and the PP, while this is not possible if the PP is associated with the object:

(53) Sarah elbowed her way quickly through the crowd.

(54) a. Sarah elbowed a friend from London quickly.

b. *Sarah elbowed a friend quickly from London.

The locus of the English *way*-construction is the word *way*, which receives the following specification:

(55) way N $(\uparrow \text{PRED}) = \text{`way'}$
$\lambda x.way(x) : (\uparrow_\sigma \text{VAR}) \multimap (\uparrow_\sigma \text{RESTR})$
(@ENGLISH-WAY((OBJ ↑) PRED FN))

According to this lexical entry, *way* contributes a semantic form ‘way’ and a standard noun meaning $\lambda x.way(x)$ on every occasion of its use, even in the *way*-construction. As we will see, our analysis equates the path specified in the ENGLISH-WAY template with the path denoted by *way*. Retaining the standard semantics for *way* allows us to provide a satisfactory analysis of modification of *way* and specification of possessors of *way* other than the subject, as discussed in Section 2.1. The relevant examples are:

(56) a. As ambassador, Chesterfield negotiated **Britain’s way** into the Treaty of Vienna in 1731.

b. In these last twenty years Richard Strauss has flamed **his meteoric way** into our ken – and out of it.

The ENGLISH-WAY constructional template appears in parentheses, since it is an optional contribution of the word *way*. Its argument is ((OBJ ↑) PRED FN): this expression uses inside-out functional uncertainty to refer to the f-structure in which *way* is an OBJ, (OBJ ↑), and passes the PRED FN of that f-structure as an argument to the template.

The definition of the ENGLISH-WAY template is:

(57) ENGLISH-WAY(PFN) :=
@TRANSITIVE-OBLIQUE(PFN)
{@TRAVERSAL-MEANS | @TRAVERSAL-MANNER}

$$
\begin{aligned}
&\lambda Y \lambda Q \lambda P \lambda x.\exists e.\exists e'.\exists z.P(e)(e') \wedge \\
&\quad theme(e') = x \wedge path(e') = z \wedge \\
&\quad\quad Q(z) \wedge z = Y(x) : \\
&[(\uparrow \text{SPEC})_\sigma \multimap \uparrow_\sigma] \multimap \\
&\quad [(((\text{OBJ} \uparrow)\ \text{OBL})_\sigma\ \text{PATH}) \multimap ((\text{OBJ} \uparrow)\ \text{OBL})_\sigma] \multimap \\
&\quad\quad [((\text{OBJ} \uparrow)_\sigma\ \text{EVENT1}) \multimap \\
&\quad\quad\quad ((\text{OBJ} \uparrow)_\sigma\ \text{EVENT2}) \multimap (\text{OBJ} \uparrow)_\sigma] \multimap \\
&\quad\quad\quad\quad (\uparrow \text{SPEC})_\sigma \multimap (\text{OBJ} \uparrow)_\sigma
\end{aligned}
$$

As shown in the template hierarchy, (33), this definition calls the TRANSITIVE-OBLIQUE template and passes in the FN of the main verb, providing the semantic form and syntactic subcategorization specification for the construction. The second line contains a disjunction: either the TRAVERSAL-MEANS or the TRAVERSAL-MANNER template is called.[15] This is because the English *way*-construction allows either a means interpretation for the construction or a manner interpretation. The TRAVERSAL-MANNER template is defined in (58) in terms of template calls to the TRAVERSAL and MANNER templates:

(58) TRAVERSAL-MANNER := @TRAVERSAL
@MANNER

The MANNER template, defined in (59), is similar to the MEANS template, defined in (41) above, in providing a meaning constructor that is a modifier (returning as its output the same type as its input). However, the MANNER specifies that a relation R is the manner by which the event e' is achieved, rather than stating that one event is the means of the other. The MANNER and MEANS templates thus have different types.

[15] Some speakers do not find the manner interpretation well-formed. Our analysis accounts for their grammars through lexical variation: the ENGLISH-WAY template in the grammars of these speakers calls only the TRAVERSAL-MEANS template.

(59) MANNER :=

$$\lambda\mathscr{P}\lambda R\lambda e'.\mathscr{P}(R)(e') \wedge manner(e') = R :$$
$$[(\uparrow_\sigma \text{ REL}) \multimap (\uparrow_\sigma \text{ EVENT2}) \multimap \uparrow_\sigma] \multimap$$
$$[(\uparrow_\sigma \text{ REL}) \multimap (\uparrow_\sigma \text{ EVENT2}) \multimap \uparrow_\sigma]$$

Besides the template calls in the first two lines, the ENGLISH-WAY template contributes the following meaning constructor:

(60)
$$\lambda Y\lambda Q\lambda P\lambda x.\exists e.\exists e'.\exists z.P(e)(e') \wedge$$
$$theme(e') = x \wedge path(e') = z \wedge$$
$$Q(z) \wedge z = Y(x) :$$
$$[(\uparrow \text{ SPEC})_\sigma \multimap \uparrow_\sigma] \multimap$$
$$[(((\text{OBJ } \uparrow) \text{ OBL})_\sigma \text{ PATH}) \multimap ((\text{OBJ } \uparrow) \text{ OBL})_\sigma] \multimap$$
$$[((\text{OBJ } \uparrow)_\sigma \text{ EVENT1}) \multimap$$
$$((\text{OBJ } \uparrow)_\sigma \text{ EVENT2}) \multimap (\text{OBJ } \uparrow)_\sigma] \multimap$$
$$(\uparrow \text{ SPEC})_\sigma \multimap (\text{OBJ } \uparrow)_\sigma$$

This meaning constructor requires:

- a meaning Y for the *way* NP, which provides additional information about the path z that is traversed;
- a meaning Q for the oblique phrase; for example (22), *Sarah elbowed her way through the crowd*, this is the meaning of *through the crowd*, which characterizes the path z;
- a meaning P, contributed by the main verb, specifying the nature of the event e and its relation to the traversal event; for *Sarah elbowed her way through the crowd*, e is required to be an elbowing event and is the means enabling the traversal event;
- a meaning x for the possessor of *way*, which plays the role of the theme of the traversal event e'.[16]

[16] As mentioned in footnote 4, the *way*-phrase may be an indefinite:

(i) Daimler bought a way into the Chinese market and then removed Chrysler from it.
`www.thetruthaboutcars.com/2013/04/`
`jeep-eyeing-chinese-cherokee-production/` (retrieved May 28, 2013)

The analysis in the body of the paper does not currently capture indefinite cases, but it could be augmented by providing, in the ENGLISH-WAY template, an optional existential closure over the term $(\uparrow \text{ SPEC})_\sigma$; this would be simi-

This analysis produces the meaning in (61) for the means interpretation of *Sarah elbowed her way through the crowd*:

(61) $\exists e.\exists e'.\exists z.elbow(e) \wedge agent(e) = sarah \wedge cause(e') = sarah \wedge$
$means(e') = e \wedge traversal(e') \wedge theme(e') = sarah \wedge$
$path(e') = z \wedge through(z, \iota x.[crowd(x)]) \wedge$
$z = \iota y.[way(y) \wedge R_c(sarah, y)]$

The main difference between this meaning and the meaning of its Swedish counterpart *Sarah armbågade sig genom mängden* is that the English *way*-construction provides a more detailed specification of the path z. We follow Partee (1983/1997) and Partee and Borschev (1998) in treating the genitive construction as involving reference to a unique individual who bears some contextually specified relation R_c to a possessor. The possessive pronoun in the phrase *her way* is resolved to the subject *Sarah*, and the meaning of *her way* is analyzed as $\iota y.[way(y) \wedge R_c(sarah, y)]$, the unique y that is a way and that bears the relation R_c to Sarah. This analysis enables us to treat cases in which *way* is modified or possessed by an individual other than the subject of the construction. Full proofs for *Sarah elbowed her way through the crowd* and *Chesterfield negotiated Britain's way into the Treaty of Vienna* are given in the Appendix.

lar to how optional objects of semantically transitive verbs like *eat* are handled in the analysis of Asudeh and Giorgolo (2012). The relevant first conjunct of example (i) would then have the meaning $\exists x'.\exists e.\exists e'.\exists z.buy(e) \wedge agent(e) = daimler \wedge cause(e') = daimler \wedge means(e') = e \wedge traversal(e') \wedge theme(e') = x' \wedge path(e') = z \wedge into(z, \iota x.[chinese\text{-}market(x)]) \wedge z = \exists y.[way(y) \wedge R_c(x', y)]$. In sum, the example would be interpreted such that Daimler bought *someone*'s way into the Chinese market (where the path traversal is interpreted metaphorically). The fact that this is likely to have been Daimler's own way is a matter of a further, bridging-like pragmatic inference (Haviland and Clark, 1974).

4.4 *Traversal constructions in Dutch*

The templates for the Dutch *weg*-construction and the Transition to Location Construction are as follows:

(62) DUTCH-WEG(PFN) :=
@DITRANSITIVE-OBLIQUE(PFN)
{@TRAVERSAL-MEANS | @TRAVERSAL-MANNER}

$$\lambda Y \lambda Q \lambda P \lambda y \lambda x.\exists e.\exists e'.\exists z.P(e)(e') \wedge theme(e') = y \wedge path(e') = z \wedge Q(z) \wedge z = Y(x) \wedge coextensive(e, e') :$$
$$[(\uparrow \text{SPEC})_\sigma \multimap \uparrow_\sigma] \multimap [(((\text{OBJ}_\theta \uparrow)\ \text{OBL})_\sigma\ \text{PATH}) \multimap ((\text{OBJ}_\theta \uparrow)\ \text{OBL})_\sigma] \multimap [((\text{OBJ}_\theta \uparrow)_\sigma\ \text{EVENT1}) \multimap ((\text{OBJ}_\theta \uparrow)_\sigma\ \text{EVENT2}) \multimap (\text{OBJ}_\theta \uparrow)_\sigma] \multimap (\uparrow \text{OBJ})_\sigma \multimap (\uparrow \text{SPEC})_\sigma \multimap (\text{OBJ}_\theta \uparrow)_\sigma$$

DUTCH-WEG, like the English *way*-construction, allows either a means or manner interpretation and further specifies that the events denoted by the main verb and the traversal event are coextensive.

(63) DUTCH-TLC(PFN) :=
@TRANSITIVE-PREDICATIVE(PFN)
@OBJ-CONTROL
@TRANSITION
@MEANS

$$\lambda Q \lambda P \lambda x.\exists e.\exists e'.P(e)(e') \wedge theme(e') = x \wedge Q(x) :$$
$$[(\uparrow \text{OBJ})_\sigma \multimap (\uparrow \text{XCOMP})_\sigma] \multimap [(\uparrow_\sigma\ \text{EVENT1}) \multimap (\uparrow_\sigma\ \text{EVENT2}) \multimap \uparrow_\sigma] \multimap (\uparrow \text{OBJ})_\sigma \multimap \uparrow_\sigma$$

DUTCH-TLC involves a transition but not necessarily a traversal, and so is defined in terms of the TRANSITION template. It specifies a means interpretation (and disallows a manner interpretation), and so incorporates the MEANS template in its definition. We assume that the prepositional phrase serves as a secondary predication on the object, since van Egmond (2006, 2009) argues that there is no path traversal in this case and also notes that the TLC bears some similarities to the resultative. In LFG-theoretic terms, this indicates that the PP is an XCOMP, which is why there is a call to OBJ-CONTROL.

5 LINKING

We now return to the definition of syntactic subcategorization requirements in the templates that appear as defaults in verbal lexical entries and as specifications of subcategorization requirements in the *way*- and DMC constructions. Recall that for simplicity, we assumed that the relation between semantic roles and grammatical functions is fixed by the construction or by information in the lexical entry of a predicate. For example, the default subcategorization for a verb like *elbowed/armbågade* was given by the TRANSITIVE template, defined above as:

(45) TRANSITIVE(PFN) := $(\uparrow \text{PRED}) = \text{`PFN}\langle(\uparrow \text{SUBJ}),(\uparrow \text{OBJ})\rangle\text{'}$

This is overly inflexible; the correct analysis would specify argument structure information for the predicate or construction rather than a specific set of grammatical functions, and would appeal to some version of Mapping Theory (Bresnan and Zaenen, 1990; Alsina, 1993; Butt, 1995; Butt *et al.*, 1997) to derive the syntactic subcategorization frame for the predicate from argument structure. We sketch here how this would work for the lexical specifications for the verb *elbow*, following the approach of Butt *et al.* (1997).

Butt *et al.* (1997) assume the following projection architecture:

(64)
$$\begin{array}{c} \text{V} \\ | \\ \text{elbow} \end{array} \xrightarrow{\alpha} \left[\begin{array}{ll} \text{REL} & \text{ELBOW} \\ \text{AGENT} & [\,] \\ \text{THEME} & [\,] \end{array}\right] \begin{array}{l} \xrightarrow{\lambda} f1 : [\,] \xrightarrow{\sigma} s1 : [\,] \\ \longrightarrow f2 : [\,] \longrightarrow s2 : [\,] \end{array}$$

Argument structure is represented as an attribute-value matrix reachable from the c-structure via the α projection. The familiar ϕ projection is defined as the composition of the α projection to argument structure and the λ projection from argument structure to f-structure.

The lexical entry for *elbowed/armbågade* can now be stated as:[17]

[17] The variable $\widehat{*}$ refers to the mother of the c-structure node that bears the annotation. Thus, $\widehat{*}_\alpha$ in the lexical entry refers to the argument structure of the mother of the terminal node *elbowed/armbågade*, i.e. the argument structure of V.

(65) elbowed/armbågade V

$$\lambda e.elbow(e) : (\uparrow_\sigma \textsc{rel})$$

$$\left(\begin{array}{l} (\uparrow \textsc{pred fn}) = \text{elbow} \\ \quad \lambda R \lambda x \lambda y \lambda e.R(e) \ \wedge \ agent(e) = x \ \wedge \ theme(e) = y : \\ \quad (\uparrow_\sigma \textsc{rel}) \multimap (\widehat{*}_\alpha \ \textsc{agent})_{\lambda\sigma} \multimap \\ \qquad (\widehat{*}_\alpha \ \textsc{theme})_{\lambda\sigma} \multimap (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma \end{array} \right)$$

Instead of specifying the default grammatical functions SUBJ and OBJ, this lexical entry specifies a default argument structure containing an AGENT and a THEME. These will be linked to the appropriate grammatical functions according to mapping theory.[18]

The English *way*-construction and the Swedish DMC construction could be treated similarly, with argument structure roles specified in the templates for the construction, and the mapping from argument structure roles to grammatical functions provided by mapping theory. However, these constructions do in fact seem to be syntactically inflexible, and cannot undergo passivization or other argument alternations:[19]

(66) *Bill's way through the park was elbowed (by him).

(67) *Bill armbågades genom parken (av sig/sig själv).
Bill elbow.PASS through park.DEF by SELF/himself

Given this, we propose to leave the templates appearing in those constructions in their current form, since we believe that specifying particular grammatical functions and disallowing argument alternations such as passive is the right treatment for these.

6 CONCLUSION

We have shown that it is not necessary to eliminate the word/phrase or lexicon/grammar distinctions in order to capture constructional effects in a principled manner. We did so by showing how this could be done in the context of Lexical-Functional Grammar, which upholds the Strong Lexicalist Hypothesis through the Lexical Integrity Princi-

[18] See Asudeh and Giorgolo (2012) for an alternative representation of argument structure.

[19] The judgement for the Swedish example is for the DMC interpretation; see Toivonen (2002) for further discussion.

ple. Like other LFG work, our approach retains Lexical Integrity as a foundational principle, due to the multifaceted nature of grammatical representation in LFG: the same functional structure and semantics can be associated with a part of a word, a word, or a phrase, but this does not imply that words and phrases are indistinguishable at other levels. We can capture the fundamental differences between words and phrases that motivate Lexical Integrity at the level of constituent structure, while also capturing commonalities in the abstract syntactic and semantic contributions of words and phrases.

Our approach captures the intuitions of Construction Grammar in an LFG setting by the use of templates, which allow for generalizations to be expressed by naming and reusing grammatical descriptions. We accomplish this without in any sense admitting constructions as first-class entities in the theory: the ability to name and reuse descriptions adds no new formal power or new formal objects to the theory. Though templates were independently motivated in much previous work for reasons of expediency in grammar writing, they now play a crucial theoretical role: templates serve as the locus of grammatical information that can be either lexically or structurally invoked, and they thus formalize one aspect of the lexicon–syntax interface. The templates are nevertheless just abbreviations for grammatical descriptions: a grammar with templates is extensionally equivalent to the same grammar with all template calls replaced by the corresponding template content. According to this view, then, in an important sense constructions are epiphenomenal.

REFERENCES

Farrell ACKERMAN and Gregory STUMP (2004), Paradigms and Periphrastic Expression: A Study in Realization-Based Lexicalism, in Louisa SADLER and Andrew SPENCER, editors, *Projecting Morphology*, pp. 111–157, CSLI Publications, Stanford, CA.

Farrell ACKERMAN and Gert WEBELHUTH (1998), *A Theory of Predicates*, CSLI Publications, Stanford, CA.

Alex ALSINA (1993), *Predicate Composition: A Theory of Syntactic Function Alternations*, Ph.D. thesis, Stanford University.

Stephen R. ANDERSON (1992), *A-Morphous Morphology*, Cambridge University Press, Cambridge, UK.

Mark ARONOFF (1993), *Morphology by Itself: Stems and Inflectional Classes*, The MIT Press, Cambridge, MA.

Ash ASUDEH (2004), *Resumption as Resource Management*, Ph.D. thesis, Stanford University.

Ash ASUDEH (2006), Direct Compositionality and the Architecture of LFG, in Miriam BUTT, Mary DALRYMPLE, and Tracy Holloway KING, editors, *Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan*, CSLI Publications, Stanford, CA.

Ash ASUDEH (2012), *The Logic of Pronominal Resumption*, Oxford University Press, Oxford.

Ash ASUDEH, Mary DALRYMPLE, and Ida TOIVONEN (2008), Constructions with Lexical Integrity: Templates as the Lexicon–Syntax Interface, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG08 Conference*, pp. 68–88, CSLI Publications, Stanford, CA.

Ash ASUDEH and Gianluca GIORGOLO (2012), Flexible Composition for Optional and Derived Arguments, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG12 Conference*, pp. 64–84, CSLI Publications, Stanford, CA.

Kenneth R. BEESLEY and Lauri KARTTUNEN (2000), Finite-State Non-Concatenative Morphotactics, in *Proceedings of the Fifth Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON-2000)*, pp. 1–12.

Kenneth R. BEESLEY and Lauri KARTTUNEN (2003), *Finite-State Morphology*, CSLI Publications, Stanford, CA.

James P. BLEVINS (2006), Word-Based Morphology, *Journal of Linguistics*, 42:531–573.

Hans C. BOAS and Ivan A. SAG, editors (2012), *Sign-Based Construction Grammar*, CSLI Publications, Stanford, CA.

Geert BOOIJ (2005a), Compounding and Derivation: Evidence for Construction Morphology, in Wolfgang U. DRESSLER, Dieter KASTOVSKY, Oskar PFEIFFER, and Franz RAINER, editors, *Morphology and its Demarcation*, pp. 109–132, John Benjamins, Amsterdam and Philadelphia.

Geert BOOIJ (2005b), Construction-Dependent Morphology, *Lingue e Linguaggio*, 12:31–46.

Geert BOOIJ (2009), Lexical Integrity as a Formal Universal: A Constructionist View, in S. SCALISE *et al.*, editors, *Universals of Language Today*, Studies in Natural Language and Linguistic Theory, pp. 83–100, Springer.

Joan BRESNAN (1998), Morphology Competes with Syntax: Explaining Typological Variation in Weak Crossover Effects, in Pilar BARBOSA, Danny

FOX, Paul HAGSTROM, Martha MCGINNIS, and David PESETSKY, editors, *Is the Best Good Enough? Proceedings from the Workshop on Optimality in Syntax*, The MIT Press, invited paper presented at the MIT Workshop on Optimality in Syntax, May 1995.

Joan BRESNAN (2001), *Lexical-Functional Syntax*, Blackwell Publishers, Oxford.

Joan BRESNAN and Sam A. MCHOMBO (1995), The Lexical Integrity Principle: Evidence from Bantu, *Natural Language and Linguistic Theory*, 13(2):181–254.

Joan BRESNAN and Annie ZAENEN (1990), Deep Unaccusativity in LFG, in Katarzyna DZIWIREK, Patrick FARRELL, and Errapel MEJÍAS-BIKANDI, editors, *Grammatical Relations: A Cross-Theoretical Perspective*, pp. 45–57, CSLI Publications, Stanford, CA.

Charles L. BUCHANAN (1918), Ornstein and Modern Music, *The Musical Quarterly*, 4(2):174–183.

Miriam BUTT (1995), *The Structure of Complex Predicates in Urdu*, Dissertations in Linguistics, CSLI Publications, Stanford, CA, revised and corrected version of 1993 Stanford University dissertation.

Miriam BUTT, Mary DALRYMPLE, and Anette FRANK (1997), An Architecture for Linking Theory in LFG, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG97 Conference*.

Miriam BUTT and Tracy Holloway KING (2005), Restriction for Morphological Valency Alternations: The Urdu Causative, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG05 Conference*.

Miriam BUTT, Tracy Holloway KING, and John T. MAXWELL (2003), Productive encoding of Urdu complex predicates in the ParGram Project, in *Proceedings of the EACL03 Workshop on Computational Linguistics for South Asian Languages: Expanding Synergies with Europe*, pp. 9–13.

Noam CHOMSKY (1965), *Aspects of the Theory of Syntax*, The MIT Press, Cambridge, MA.

Dick CROUCH, Mary DALRYMPLE, Ron KAPLAN, Tracy KING, John MAXWELL, and Paula NEWMAN (2012), *XLE Documentation*, Palo Alto Research Center, Palo Alto, CA, URL `http://www2.parc.com/isl/groups/nltt/xle/doc/xle.html`.

Peter W. CULICOVER and Ray JACKENDOFF (1999), The View from the Periphery: The English Comparative Correlative, *Linguistic Inquiry*, 30(4):543–572.

Mary DALRYMPLE, editor (1999), *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*, The MIT Press, Cambridge, MA.

Mary DALRYMPLE (2001), *Lexical Functional Grammar*, volume 34 of *Syntax and Semantics*, Academic Press, New York, NY.

Mary DALRYMPLE, Ronald M. KAPLAN, and Tracy Holloway KING (2004), Linguistic Generalizations over Descriptions, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG04 Conference*.

Mary DALRYMPLE, Ronald M. KAPLAN, John T. MAXWELL, III, and Annie ZAENEN, editors (1995), *Formal Issues in Lexical-Functional Grammar*, CSLI Publications, Stanford, CA.

Anna Maria DI SCIULLO and Edwin WILLIAMS (1987), *On the Definition of Word*, The MIT Press, Cambridge, MA.

Yehuda N. FALK (2001), *Lexical-Functional Grammar: An Introduction to Parallel Constraint-Based Syntax*, CSLI Publications, Stanford, CA.

Charles FILLMORE (1988), The Mechanisms of 'Construction Grammar', in Shelley AXMAKER, Annie JAISSER, and Helen SINGMASTER, editors, *Proceedings of Fourteenth Annual Meeting of the Berkeley Linguistics Society*, pp. 35–55, Berkeley Linguistics Society, Berkeley.

Charles J. FILLMORE, Paul KAY, and Mary Catherine O'CONNOR (1988), Regularity and Idiomaticity in Grammatical Constructions: The Case of *Let Alone*, *Language*, 64(3):501–538.

Jonathan GINZBURG and Ivan A. SAG (2000), *Interrogative Investigations: The Form, Meaning and Use of English Interrogatives*, CSLI Publications, Stanford, CA.

Adele E. GOLDBERG (1995), *Constructions: A Construction Grammar Approach to Argument Structure*, University of Chicago Press, Chicago.

Adele E. GOLDBERG (2006), *Constructions at Work: The Nature of Generalization in Language*, Oxford University Press, Oxford.

Adele E. GOLDBERG and Ray JACKENDOFF (2004), The English Resultative as a Family of Constructions, *Language*, 80(3):532–568.

Morris HALLE and Alec MARANTZ (1993), Distributed Morphology and the Pieces of Inflection, in *The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger*, pp. 111–176, The MIT Press, Cambridge, MA.

Morris HALLE and Alec MARANTZ (1994), Some Key Features of Distributed Morphology, in Andrew CARNIE and Heidi HARLEY, editors, *MIT Working Papers in Linguistics 21: Papers on Phonology and Morphology*, pp. 275–288, Department of Linguistics and Philosophy, MIT, Cambridge, MA.

Susan E. HAVILAND and Herbert H. CLARK (1974), What's New? Acquiring New Information as a Process in Comprehension, *Journal of Verbal Learning and Verbal Behavior*, 13(5):512–521.

Ray JACKENDOFF (1973), The Base Rules for Prepositional Phrases, in Stephen ANDERSON and Paul KIPARSKY, editors, *A Festschrift for Morris Halle*, pp. 345–356, Holt, Rinehart and Winston, New York, NY.

Ray JACKENDOFF (1992), Babe Ruth Homered His Way into the Hearts of America, in Stowell and Wehrli (1992), pp. 155–178.

Ray S. JACKENDOFF (1990), *Semantic Structures*, The MIT Press, Cambridge, MA.

Ray S. JACKENDOFF (2007), *Language, Consciousness, Culture: Essays on Mental Structure*, The MIT Press, Cambridge, MA.

Theo M. V. JANSSEN (1997), Compositionality, in *Handbook of Logic and Language*, pp. 419–473, The MIT Press/Elsevier, Cambridge and Amsterdam.

Aravind K. JOSHI, K. VIJAY-SHANKER, and David WEIR (1991), The Convergence of Mildly Context-Sensitive Grammar Formalisms, in Peter SELLS, Stuart SHIEBER, and Thomas WASOW, editors, *Foundational Issues in Natural Language Processing*, pp. 31–81, MIT Press.

Ronald M. KAPLAN (1987), Three Seductions of Computational Psycholinguistics, in Peter WHITELOCK, Mary McGee WOOD, Harold L. SOMERS, Rod JOHNSON, and Paul BENNETT, editors, *Linguistic Theory and Computer Applications*, pp. 149–188, Academic Press, London, also published as CCL/UMIST Report No. 86.2: Alvey/ICL Workshop on Linguistic Theory and Computer Applications: Transcripts of Presentations and Discussions. Center for Computational Linguistics, University of Manchester. Reprinted in Dalrymple *et al.* (1995, 337–367).

Ronald M. KAPLAN and Jürgen WEDEKIND (1993), Restriction and Correspondence-Based Translation, in *Proceedings of the 6th Meeting of the EACL,* Utrecht, European Chapter of the Association for Computational Linguistics.

Paul KAY and Charles J. FILLMORE (1999), Grammatical Constructions and Linguistic Generalizations: The *What's X doing Y?* Construction, *Language*, 75(1):1–33.

Robert MALOUF (1998), Categories, Prototypes, and Default Inheritance, in *Proceedings of the Joint Conference on Formal Grammar, Head-Driven Phrase Structure Grammar, and Categorial Grammar*, pp. 207–216, Saarbrücken.

Alec MARANTZ (1992), The *Way*-Construction and the Semantics of Direct Arguments in English: A Reply to Jackendoff, in Stowell and Wehrli (1992), pp. 179–188.

Alec P. MARANTZ (1997), No Escape from Syntax: Don't Try Morphological Analysis in the Privacy of Your Own Lexicon, in A. DIMITRIADIS and L. SIEGEL, editors, *University of Pennsylvania Working Papers in Linguistics*, volume 4.2, pp. 201–226, University of Pennsylvania.

Yo MATSUMOTO (1996), *Complex Predicates in Japanese: A Syntactic and Semantic Study of the Notion 'Word'*, Ph.D. thesis, Stanford University, *Studies in Japanese Linguistics* series, CSLI Publications/Kuroiso Publishers, Stanford/Tokyo. Revised and corrected version of 1992 Stanford University dissertation, *On the wordhood of complex predicates in Japanese*.

Laura MICHAELIS (2010), Sign-Based Construction Grammar, in Bernd HEINE and Heiko NARROG, editors, *The Oxford Handbook of Linguistic Analysis*, Oxford University Press, Oxford.

Tara MOHANAN (1994), *Arguments in Hindi*, Dissertations in Linguistics, CSLI Publications, Stanford, CA, reprinted version of 1990 Stanford University dissertation.

Tara MOHANAN (1995), Wordhood and Lexicality: Noun Incorporation in Hindi, *Natural Language and Linguistic Theory*, 13(1):75–134.

Barbara PARTEE (1983/1997), Genitives – A Case Study, in Johan van BENTHEM and Alice TER MEULEN, editors, *Handbook of Logic and Language*, pp. 464–470, Elsevier/The MIT Press, Amsterdam and Boston, appendix to Janssen (1997). Published version of 1983 manuscript.

Barbara PARTEE and Vladimir BORSCHEV (1998), Integrating lexical and formal semantics: Genitives, relational nouns, and type-shifting, in Robin COOPER and Thomas GAMKRELIDZE, editors, *Proceedings of the Second Tbilisi Symposium on Language, Logic, and Computation*, pp. 229–241, Center on Language, Logic, Speech, Tbilisi State University, Tbilisi, URL `http://www-unix.oit.umass.edu/~partee/docs/tbilisi17.pdf`.

Barbara H. PARTEE, Alice TER MEULEN, and Robert E. WALL (1993), *Mathematical Methods in Linguistics*, Kluwer, Dordrecht, corrected second printing of the first edition.

Carl POLLARD and Ivan A. SAG (1987), *Information-Based Syntax and Semantics, Volume I*, CSLI Lecture Notes, number 13, CSLI Publications, Stanford, CA.

Carl POLLARD and Ivan A. SAG (1994), *Head-Driven Phrase Structure Grammar*, University of Chicago Press, Chicago.

Geoffrey K. PULLUM and Gerald GAZDAR (1982), Natural Languages and Context Free Languages, *Linguistics and Philosophy*, 4:471–504.

Louisa SADLER and Andrew SPENCER (2000), Syntax as an Exponent of Morphological Features, *Yearbook of Morphology*, pp. 71–96.

Ivan A. SAG (2010), English Filler-Gap Constructions, *Language*, 86(3):486–545.

Ivan A. SAG (2012), Sign-Based Construction Grammar: An Informal Synopsis, in Boas and Sag (2012), pp. 69–202.

Ivan A. SAG, Hans C. BOAS, and Paul KAY (2012), Introducing Sign-Based Construction Grammar, in Boas and Sag (2012), pp. 1–30.

Stuart M. SHIEBER (1985), Evidence Against the Context-Freeness of Natural Language, *Linguistics and Philosophy*, 8:333–343.

Daniel SIDDIQI (2009), *Syntax Within the Word: Economy, Allomorphy, and Argument Selection in Distributed Morphology*, John Benjamins.

Jane SIMPSON (1983), *Aspects of Warlpiri Morphology and Syntax*, Ph.D. thesis, MIT.

Tim STOWELL and Eric WEHRLI, editors (1992), *Syntax and the Lexicon*, volume 26 of *Syntax and Semantics*, Academic Press, San Diego, CA.

Gregory T. STUMP (2001), *Inflectional Morphology: A Theory of Paradigm Structure*, Cambridge University Press, Cambridge, UK.

Ida TOIVONEN (2002), The Directed Motion Construction in Swedish, *Journal of Linguistics*, 38(2):313–345.

Ida TOIVONEN (2003), *Non-Projecting Words: A Case Study of Swedish Particles*, Kluwer Academic Publishers, Dordrecht.

Marie-Elaine VAN EGMOND (2006), *Two* Way-*Constructions in Dutch*, Master's thesis, University of Canterbury, Christchurch, New Zealand.

Marie-Elaine VAN EGMOND (2009), *Two* Way-*Constructions in Dutch: Motion Along a Path and Transition to Location*, VDM Verlag Dr. Müller, Saarbrücken, Germany.

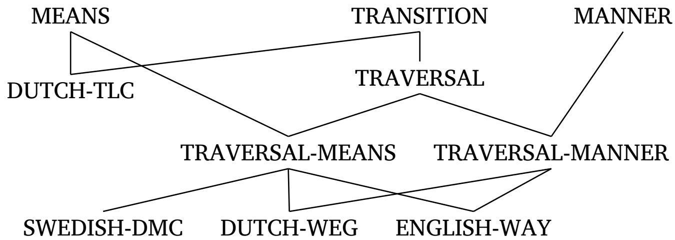
Arie VERHAGEN (2003), The Dutch Way, in Arie VERHAGEN and Jeroen M. van de WEIJER, editors, *Usage-Based Approaches to Dutch*, pp. 27–57, LOT, Utrecht.

K. VIJAY-SHANKER and David WEIR (1994), The Equivalence of Four Extensions of Context-Free Grammars, *Mathematical Systems Theory*, 27(6):511–546.

APPENDIX

A TEMPLATE HIERARCHY

(68)



B TEMPLATES

(69) TRANSITION :=

$\lambda R \lambda x \lambda e \lambda e'.R(e) \;\wedge\; agent(e) = x \;\wedge\; cause(e') = x :$

$(\uparrow_\sigma \textsc{rel}) \multimap (\uparrow \textsc{subj})_\sigma \multimap (\uparrow_\sigma \textsc{event1}) \multimap (\uparrow_\sigma \textsc{event2}) \multimap \uparrow_\sigma$

(70) MEANS :=

$\lambda P \lambda e \lambda e'.P(e)(e') \;\wedge\; means(e') = e :$

$[(\uparrow_\sigma \textsc{event1}) \multimap (\uparrow_\sigma \textsc{event2}) \multimap \uparrow_\sigma] \multimap$
$[(\uparrow_\sigma \textsc{event1}) \multimap (\uparrow_\sigma \textsc{event2}) \multimap \uparrow_\sigma]$

(71) MANNER :=

$\lambda \mathscr{P} \lambda R \lambda e'.\mathscr{P}(R)(e') \;\wedge\; manner(e') = R :$

$[(\uparrow_\sigma \textsc{rel}) \multimap (\uparrow_\sigma \textsc{event2}) \multimap \uparrow_\sigma] \multimap$
$[(\uparrow_\sigma \textsc{rel}) \multimap (\uparrow_\sigma \textsc{event2}) \multimap \uparrow_\sigma]$

(72) TRAVERSAL :=

@TRANSITION

$\lambda P \lambda e'.P(e') \;\wedge\; traversal(e') :$

$[(\uparrow_\sigma \textsc{event2}) \multimap \uparrow_\sigma] \multimap [(\uparrow_\sigma \textsc{event2}) \multimap \uparrow_\sigma]$

(73) TRAVERSAL-MEANS := @TRAVERSAL
@MEANS

(74) TRAVERSAL-MANNER := @TRAVERSAL
@MANNER

(75) SWEDISH-DMC(PFN) :=

@TRANSITIVE-OBLIQUE(PFN)

@TRAVERSAL-MEANS

$\lambda Q \lambda P \lambda y.\exists e.\exists e'.\exists z.P(e)(e') \;\wedge$
$theme(e') = y \;\wedge\; path(e') = z \;\wedge\; Q(z) :$

$[((\uparrow \textsc{obl})_\sigma \textsc{path}) \multimap (\uparrow \textsc{obl})_\sigma] \multimap$
$[(\uparrow_\sigma \textsc{event1}) \multimap (\uparrow_\sigma \textsc{event2}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow \textsc{obj})_\sigma \multimap \uparrow_\sigma$

(76) ENGLISH-WAY(PFN) :=
@TRANSITIVE-OBLIQUE(PFN)
{@TRAVERSAL-MEANS | @TRAVERSAL-MANNER}

$$\lambda Y \lambda Q \lambda P \lambda x.\exists e.\exists e'.\exists z.P(e)(e') \wedge theme(e') = x \wedge path(e') = z \wedge Q(z) \wedge z = Y(x) :$$

$$[(\uparrow \textsc{spec})_\sigma \multimap \uparrow_\sigma] \multimap [(((\textsc{obj}\ \uparrow)\ \textsc{obl})_\sigma\ \textsc{path}) \multimap ((\textsc{obj}\ \uparrow)\ \textsc{obl})_\sigma] \multimap [((\textsc{obj}\ \uparrow)_\sigma\ \textsc{event1}) \multimap ((\textsc{obj}\ \uparrow)_\sigma\ \textsc{event2}) \multimap (\textsc{obj}\ \uparrow)_\sigma] \multimap (\uparrow \textsc{spec})_\sigma \multimap (\textsc{obj}\ \uparrow)_\sigma$$

(77) DUTCH-WEG(PFN) :=
@DITRANSITIVE-OBLIQUE(PFN)
{@TRAVERSAL-MEANS | @TRAVERSAL-MANNER}

$$\lambda Y \lambda Q \lambda P \lambda y \lambda x.\exists e.\exists e'.\exists z.P(e)(e') \wedge theme(e') = y \wedge path(e') = z \wedge Q(z) \wedge z = Y(x) \wedge coextensive(e, e') :$$

$$[(\uparrow \textsc{spec})_\sigma \multimap \uparrow_\sigma] \multimap [(((\textsc{obj}_\theta\ \uparrow)\ \textsc{obl})_\sigma\ \textsc{path}) \multimap ((\textsc{obj}_\theta\ \uparrow)\ \textsc{obl})_\sigma] \multimap [((\textsc{obj}_\theta\ \uparrow)_\sigma\ \textsc{event1}) \multimap ((\textsc{obj}_\theta\ \uparrow)_\sigma\ \textsc{event2}) \multimap (\textsc{obj}_\theta\ \uparrow)_\sigma] \multimap (\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{spec})_\sigma \multimap (\textsc{obj}_\theta\ \uparrow)_\sigma$$

(78) DUTCH-TLC(PFN) :=
@TRANSITIVE-PREDICATIVE(PFN)
@OBJ-CONTROL
@TRANSITION
@MEANS

$$\lambda Q \lambda P \lambda x.\exists e.\exists e'.P(e)(e') \wedge theme(e') = x \wedge Q(x) :$$

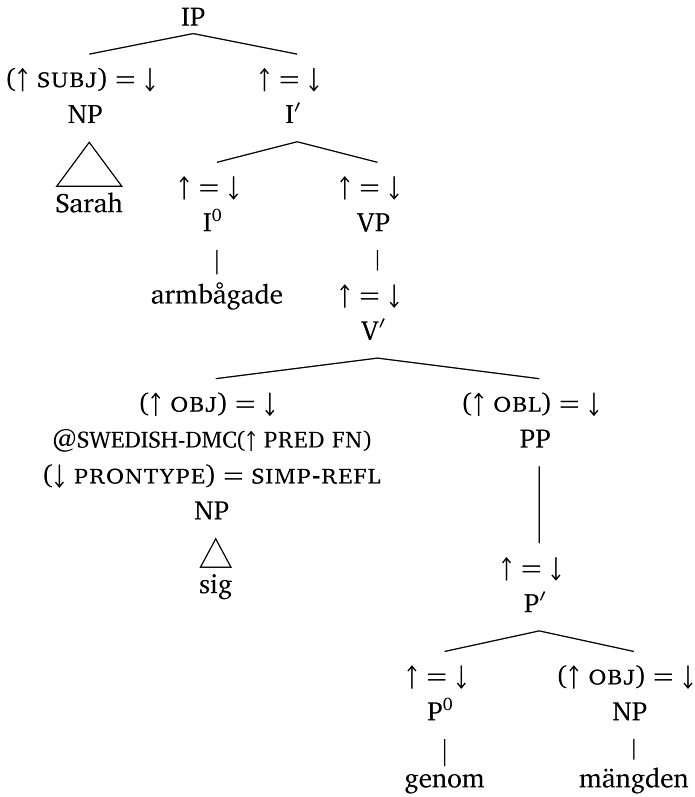
$$[(\uparrow \textsc{obj})_\sigma \multimap (\uparrow \textsc{xcomp})_\sigma] \multimap [(\uparrow_\sigma\ \textsc{event1}) \multimap (\uparrow_\sigma\ \textsc{event2}) \multimap \uparrow_\sigma] \multimap (\uparrow \textsc{obj})_\sigma \multimap \uparrow_\sigma$$

C EXAMPLES

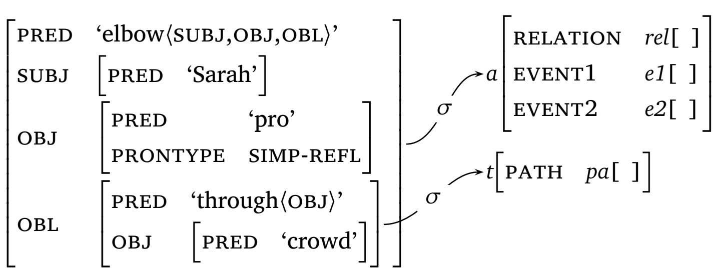
C.1 *Swedish*

(79) Sarah armbågade sig genom mängden.
S. elbowed SELF through crowd.DEF

(80)



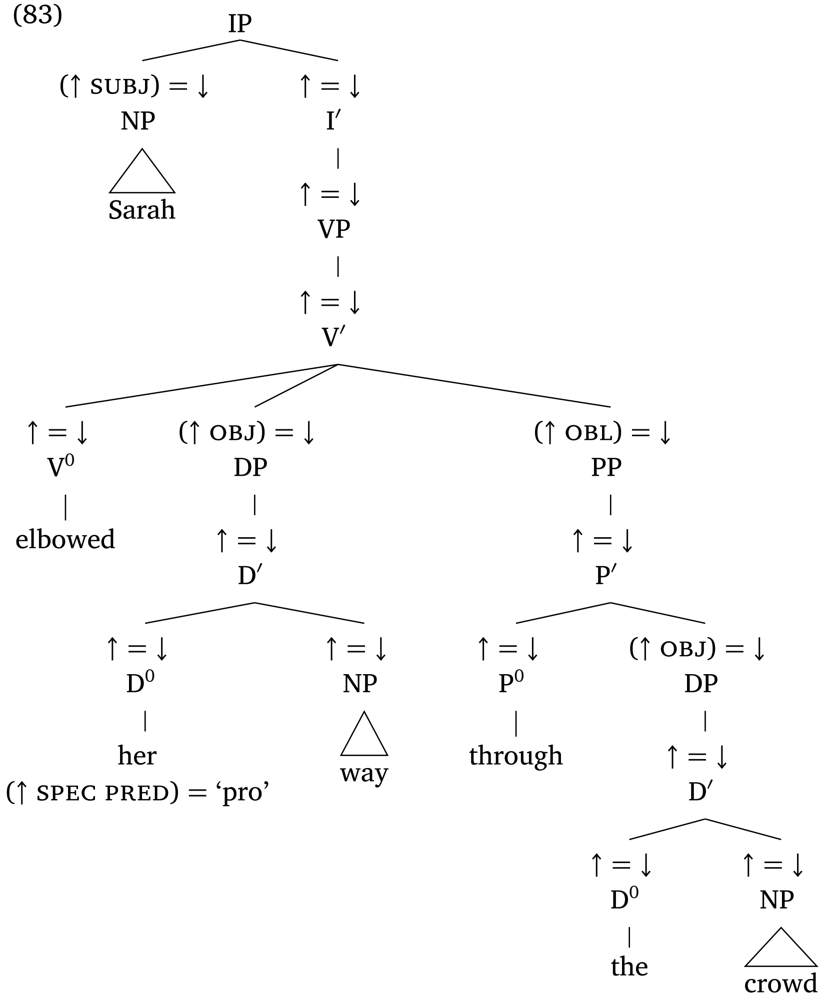
(81)



C.2 *English*

(82) Sarah elbowed her way through the crowd.

(83)



(84)

$$
\left[\begin{array}{ll}
\textsc{pred} & \text{‘elbow}\langle\textsc{subj},\textsc{obj},\textsc{obl}\rangle\text{’} \\
\textsc{subj} & \left[\begin{array}{ll}\textsc{pred} & \text{‘Sarah’}\end{array}\right] \\
\textsc{obj} & \left[\begin{array}{ll}\textsc{pred} & \text{‘way’} \\ \textsc{spec} & \left[\begin{array}{ll}\textsc{pred} & \text{‘pro’}\end{array}\right]\end{array}\right] \\
\textsc{obl} & \left[\begin{array}{ll}\textsc{pred} & \text{‘through}\langle\textsc{obj}\rangle\text{’} \\ \textsc{obj} & \left[\begin{array}{ll}\textsc{pred} & \text{‘crowd’}\end{array}\right]\end{array}\right]
\end{array}\right]
\begin{array}{l}
\xrightarrow{\sigma} a\left[\begin{array}{ll}\textsc{relation} & \mathit{rel}[\ ] \\ \textsc{event1} & \mathit{e1}[\ ] \\ \textsc{event2} & \mathit{e2}[\ ]\end{array}\right] \\
\xrightarrow{\sigma} w\left[\begin{array}{ll}\textsc{var} & v[\ ] \\ \textsc{restr} & r[\ ]\end{array}\right] \\
\xrightarrow{\sigma} t\left[\begin{array}{ll}\textsc{path} & \mathit{pa}[\ ]\end{array}\right]
\end{array}
$$

armbågade

$$\lambda e.elbow(e) : rel$$

Transition

$$\lambda R\lambda x\lambda e\lambda e'.R(e) \wedge agent(e) = x \wedge cause(e') = x : rel \multimap s \multimap e1 \multimap e2 \multimap a$$

$$\lambda x\lambda e\lambda e'.elbow(e) \wedge agent(e) = x \wedge cause(e') = x : s \multimap e1 \multimap e2 \multimap a$$

$$[y_1 : s]^1$$

$$\lambda e\lambda e'.elbow(e) \wedge agent(e) = y_1 \wedge cause(e') = y_1 : e1 \multimap e2 \multimap a$$

Means

$$\lambda P\lambda e\lambda e'.P(e)(e') \wedge means(e') = e : (e1 \multimap e2 \multimap a) \multimap (e1 \multimap e2 \multimap a)$$

$$\lambda e\lambda e'.elbow(e) \wedge agent(e) = y_1 \wedge cause(e') = y_1 \wedge means(e') = e : e1 \multimap e2 \multimap a$$

$$[e'' : e1]^3$$

$$\lambda e'.elbow(e'') \wedge agent(e'') = y_1 \wedge cause(e') = y_1 \wedge means(e') = e'' : e2 \multimap a$$

Traversal

$$\lambda P\lambda e'.P(e') \wedge traversal(e') : (e2 \multimap a) \multimap (e2 \multimap a)$$

$$\lambda e'.elbow(e'') \wedge agent(e'') = y_1 \wedge cause(e') = y_1 \wedge means(e') = e'' \wedge traversal(e') : e2 \multimap a$$

$$\lambda e''\lambda e'.elbow(e'') \wedge agent(e'') = y_1 \wedge cause(e') = y_1 \wedge means(e') = e'' \wedge traversal(e') : e1 \multimap e2 \multimap a \quad \multimap_{\mathcal{I},3}$$

Swedish-DMC

$$\lambda Q\lambda P\lambda y.\exists e.\exists e'.\exists z.P(e)(e') \wedge theme(e') = y \wedge path(e') = z \wedge Q(z) : (pa \multimap t) \multimap (e1 \multimap e2 \multimap a) \multimap p \multimap a$$

genom

$$\lambda z\lambda y.through(y,z) : m \multimap pa \multimap t$$

mängden

$$\iota x.[crowd(x)] : m$$

$$\lambda y.through(y, \iota x.[crowd(x)]) : pa \multimap t$$

$$\lambda P\lambda y.\exists e.\exists e'.\exists z.P(e)(e') \wedge theme(e') = y \wedge path(e') = z \wedge through(z, \iota x.[crowd(x)]) : (e1 \multimap e2 \multimap a) \multimap p \multimap a$$

$$\lambda y.\exists e.\exists e'.\exists z.elbow(e) \wedge agent(e) = y_1 \wedge cause(e') = y_1 \wedge means(e') = e \wedge traversal(e') \wedge theme(e') = y \wedge path(e') = z \wedge through(z, \iota x.[crowd(x)]) : p \multimap a$$

$$[x_1 : p]^2$$

$$\exists e.\exists e'.\exists z.elbow(e) \wedge agent(e) = y_1 \wedge cause(e') = y_1 \wedge means(e') = e \wedge traversal(e') \wedge theme(e') = x_1 \wedge path(e') = z \wedge through(z, \iota x.[crowd(x)]) : a$$

Sarah

$$sarah : s$$

sig

$$\lambda y.y \times y : s \multimap s \otimes p$$

$$sarah \times sarah : s \otimes p$$

$$\exists e.\exists e'.\exists z.elbow(e) \wedge agent(e) = sarah \wedge cause(e') = sarah \wedge means(e') = e \wedge traversal(e') \wedge theme(e') = sarah \wedge path(e') = z \wedge through(z, \iota x.[crowd(x)]) : a \quad \otimes_{\mathcal{E},1,2}$$

Figure 1: Glue Semantics proof for (79); Swedish Directed Motion Construction

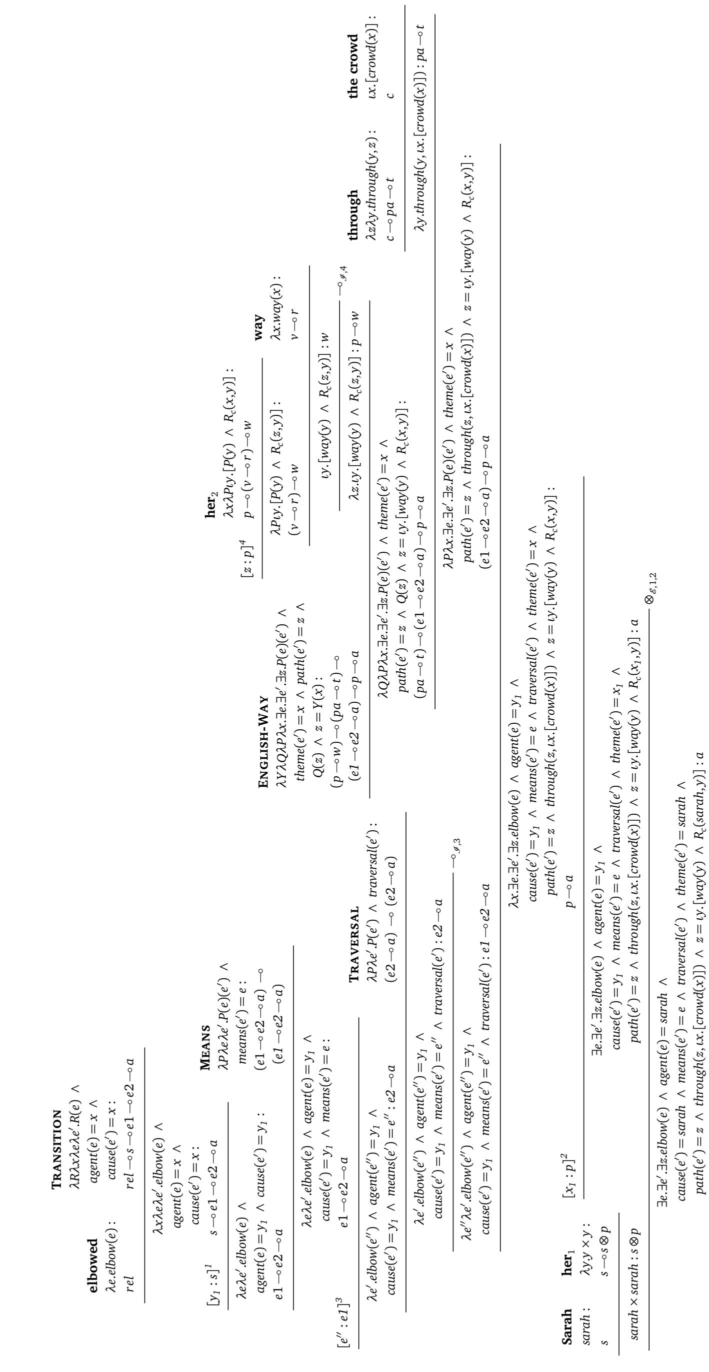


Figure 2: Glue Semantics proof for (82); English *Way* Construction (means interpretation)

$$\dfrac{[R' : rel]^4 \qquad \begin{array}{l}\textbf{Transition}\\ \lambda R \lambda x \lambda e \lambda e'.R(e) \wedge \\ \quad agent(e) = x \wedge \\ \quad cause(e') = x : \\ rel \multimap s \multimap e1 \multimap e2 \multimap a\end{array}}{\begin{array}{l}\lambda x \lambda e \lambda e'.R'(e) \wedge \\ \quad agent(e) = x \wedge \\ \quad cause(e') = x : \\ s \multimap e1 \multimap e2 \multimap a\end{array}}$$

$$\dfrac{[y_1 : s]^1 \qquad \lambda x \lambda e \lambda e'.R'(e) \wedge agent(e) = x \wedge cause(e') = x : s \multimap e1 \multimap e2 \multimap a}{\begin{array}{l}\lambda e \lambda e'.R'(e) \wedge \\ \quad agent(e) = y_1 \wedge \\ \quad cause(e') = y_1 : \\ e1 \multimap e2 \multimap a\end{array}}$$

$$\dfrac{[e'' : e1]^3 \qquad \lambda e \lambda e'.R'(e) \wedge agent(e) = y_1 \wedge cause(e') = y_1 : e1 \multimap e2 \multimap a}{\begin{array}{l}\lambda e'.R'(e'') \wedge agent(e'') = y_1 \wedge \\ \quad cause(e') = y_1 : \\ e2 \multimap a\end{array}}$$

$$\dfrac{\lambda e'.R'(e'') \wedge agent(e'') = y_1 \wedge cause(e') = y_1 : e2 \multimap a}{\begin{array}{l}\lambda R' \lambda e'.R'(e'') \wedge agent(e'') = y_1 \wedge \\ \quad cause(e') = y_1 : \\ rel \multimap e2 \multimap a\end{array}} \multimap_{\mathcal{I},4}$$

$$\dfrac{\lambda R' \lambda e'.R'(e'') \wedge agent(e'') = y_1 \wedge cause(e') = y_1 : rel \multimap e2 \multimap a \qquad \begin{array}{l}\textbf{Manner}\\ \lambda \mathscr{P} \lambda R \lambda e.\mathscr{P}(R)(e) \wedge \\ \quad manner(e) = R : \\ (rel \multimap e2 \multimap a) \multimap \\ \quad (rel \multimap e2 \multimap a)\end{array}}{\begin{array}{l}\lambda R \lambda e.R(e'') \wedge agent(e'') = y_1 \wedge \\ \quad cause(e) = y_1 \wedge manner(e) = R : \\ rel \multimap e2 \multimap a\end{array}}$$

$$\dfrac{\begin{array}{l}\textbf{elbowed}\\ \lambda e.elbow(e) : \\ rel\end{array} \qquad \lambda R \lambda e.R(e'') \wedge agent(e'') = y_1 \wedge cause(e) = y_1 \wedge manner(e) = R : rel \multimap e2 \multimap a}{\begin{array}{l}\lambda e.elbow(e'') \wedge agent(e'') = y_1 \wedge \\ \quad cause(e) = y_1 \wedge manner(e) = elbow : e2 \multimap a\end{array}}$$

$$\dfrac{\lambda e.elbow(e'') \wedge agent(e'') = y_1 \wedge cause(e) = y_1 \wedge manner(e) = elbow : e2 \multimap a \qquad \begin{array}{l}\textbf{Traversal}\\ \lambda P \lambda e'''.P(e''') \wedge traversal(e''') : \\ (e2 \multimap a) \multimap (e2 \multimap a)\end{array}}{\begin{array}{l}\lambda e'''.elbow(e'') \wedge agent(e'') = y_1 \wedge \\ \quad cause(e''') = y_1 \wedge manner(e''') = elbow \wedge traversal(e''') : e2 \multimap a\end{array}}$$

$$\dfrac{\lambda e'''.elbow(e'') \wedge agent(e'') = y_1 \wedge cause(e''') = y_1 \wedge manner(e''') = elbow \wedge traversal(e''') : e2 \multimap a}{\begin{array}{l}\lambda e'' \lambda e'''.elbow(e'') \wedge agent(e'') = y_1 \wedge \\ \quad cause(e''') = y_1 \wedge manner(e''') = elbow \wedge traversal(e''') : e1 \multimap e2 \multimap a\end{array}} \multimap_{\mathcal{I},3}$$

$$\dfrac{[z : p]^5 \qquad \begin{array}{l}\textbf{her}_2\\ \lambda x \lambda P \iota y.[P(y) \wedge R_c(x,y)] : \\ p \multimap (v \multimap r) \multimap w\end{array}}{\begin{array}{l}\lambda P \iota y.[P(y) \wedge R_c(z,y)] : \\ (v \multimap r) \multimap w\end{array}}$$

$$\dfrac{\lambda P \iota y.[P(y) \wedge R_c(z,y)] : (v \multimap r) \multimap w \qquad \begin{array}{l}\textbf{way}\\ \lambda x.way(x) : \\ v \multimap r\end{array}}{\iota y.[way(y) \wedge R_c(z,y)] : w}$$

$$\dfrac{\iota y.[way(y) \wedge R_c(z,y)] : w}{\lambda z.\iota y.[way(y) \wedge R_c(z,y)] : p \multimap w} \multimap_{\mathcal{I},5}$$

$$\dfrac{\begin{array}{l}\textbf{English-Way}\\ \lambda Y \lambda Q \lambda P \lambda x.\exists e.\exists e'.\exists z.P(e)(e') \wedge \\ \quad theme(e') = x \wedge path(e') = z \wedge \\ \quad Q(z) \wedge z = Y(x) : \\ (p \multimap w) \multimap (pa \multimap t) \multimap \\ \quad (e1 \multimap e2 \multimap a) \multimap p \multimap a\end{array} \qquad \lambda z.\iota y.[way(y) \wedge R_c(z,y)] : p \multimap w}{\begin{array}{l}\lambda Q \lambda P \lambda x.\exists e.\exists e'.\exists z.P(e)(e') \wedge theme(e') = x \wedge \\ \quad path(e') = z \wedge Q(z) \wedge z = \iota y.[way(y) \wedge R_c(x,y)] : \\ (pa \multimap t) \multimap (e1 \multimap e2 \multimap a) \multimap p \multimap a\end{array}}$$

$$\dfrac{\begin{array}{l}\textbf{through}\\ \lambda z \lambda y.through(y,z) : \\ c \multimap pa \multimap t\end{array} \qquad \begin{array}{l}\textbf{the crowd}\\ \iota x.[crowd(x)] : \\ c\end{array}}{\lambda y.through(y, \iota x.[crowd(x)]) : pa \multimap t}$$

$$\dfrac{\begin{array}{l}\lambda Q \lambda P \lambda x.\exists e.\exists e'.\exists z.P(e)(e') \wedge theme(e') = x \wedge \\ \quad path(e') = z \wedge Q(z) \wedge z = \iota y.[way(y) \wedge R_c(x,y)] : \\ (pa \multimap t) \multimap (e1 \multimap e2 \multimap a) \multimap p \multimap a\end{array} \qquad \lambda y.through(y, \iota x.[crowd(x)]) : pa \multimap t}{\begin{array}{l}\lambda P \lambda x.\exists e.\exists e'.\exists z.P(e)(e') \wedge theme(e') = x \wedge \\ \quad path(e') = z \wedge through(z, \iota x.[crowd(x)]) \wedge z = \iota y.[way(y) \wedge R_c(x,y)] : \\ (e1 \multimap e2 \multimap a) \multimap p \multimap a\end{array}}$$

$$\dfrac{\lambda e'' \lambda e'''.elbow(e'') \wedge \ldots : e1 \multimap e2 \multimap a \qquad \lambda P \lambda x.\exists e.\exists e'.\exists z.P(e)(e') \wedge \ldots : (e1 \multimap e2 \multimap a) \multimap p \multimap a}{\begin{array}{l}\lambda x.\exists e.\exists e'.\exists z.elbow(e) \wedge agent(e) = y_1 \wedge \\ \quad cause(e') = y_1 \wedge manner(e') = elbow \wedge traversal(e') \wedge theme(e') = x \wedge \\ \quad path(e') = z \wedge through(z, \iota x.[crowd(x)]) \wedge z = \iota y.[way(y) \wedge R_c(x,y)] : \\ p \multimap a\end{array}}$$

$$\dfrac{[x_1 : p]^2 \qquad \lambda x.\exists e.\exists e'.\exists z.elbow(e) \wedge \ldots : p \multimap a}{\begin{array}{l}\exists e.\exists e'.\exists z.elbow(e) \wedge agent(e) = y_1 \wedge \\ \quad cause(e') = y_1 \wedge manner(e') = elbow \wedge traversal(e') \wedge theme(e') = x_1 \wedge \\ \quad path(e') = z \wedge through(z, \iota x.[crowd(x)]) \wedge z = \iota y.[way(y) \wedge R_c(x_1,y)] : a\end{array}}$$

$$\dfrac{\begin{array}{l}\textbf{Sarah}\\ sarah : \\ s\end{array} \qquad \begin{array}{l}\textbf{her}_1\\ \lambda y.y \times y : \\ s \multimap s \otimes p\end{array}}{sarah \times sarah : s \otimes p}$$

$$\dfrac{sarah \times sarah : s \otimes p \qquad \exists e.\exists e'.\exists z.elbow(e) \wedge \ldots : a}{\begin{array}{l}\exists e.\exists e'.\exists z.elbow(e) \wedge agent(e) = sarah \wedge \\ \quad cause(e') = sarah \wedge manner(e') = elbow \wedge traversal(e') \wedge theme(e') = sarah \wedge \\ \quad path(e') = z \wedge through(z, \iota x.[crowd(x)]) \wedge z = \iota y.[way(y) \wedge R_c(sarah,y)] : a\end{array}} \otimes_{\mathcal{E},1,2}$$

Figure 3: Glue Semantics proof for (82); English *Way* Construction (manner interpretation)

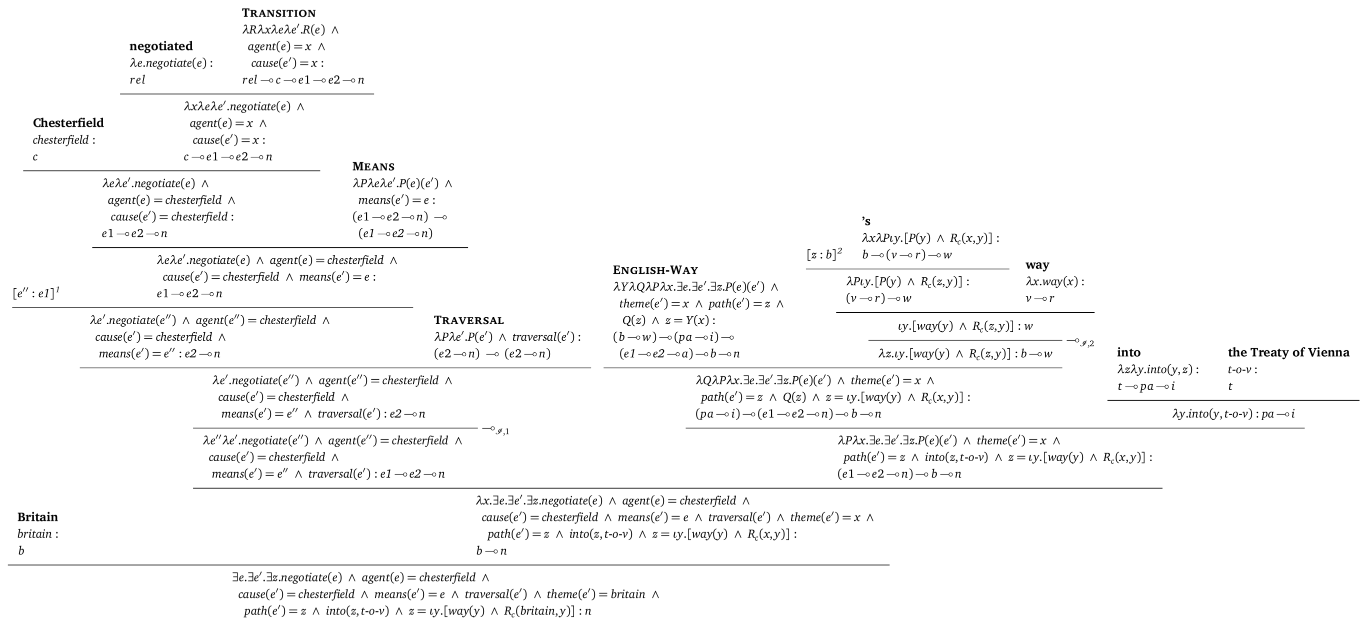


Figure 4: Glue Semantics proof for *Chesterfield negotiated Britain’s way into the Treaty of Vienna*; English *Way* Construction (means interpretation)

This work is licensed under the Creative Commons Attribution 3.0 Unported License.
http://creativecommons.org/licenses/by/3.0/

Text: now in 2D!
A framework for lexical expansion with contextual similarity

Chris Biemann and Martin Riedl
Computer Science Department, FG Language Technology,
TU Darmstadt, Germany
{biem,riedl}@cs.tu-darmstadt.de

ABSTRACT

A new metaphor of two-dimensional text for data-driven semantic modeling of natural language is proposed, which provides an entirely new angle on the representation of text: not only syntagmatic relations are annotated in the text, but also paradigmatic relations are made explicit by generating lexical expansions. We operationalize distributional similarity in a general framework for large corpora, and describe a new method to generate similar terms in context. Our evaluation shows that distributional similarity is able to produce high-quality lexical resources in an unsupervised and knowledge-free way, and that our highly scalable similarity measure yields better scores in a WordNet-based evaluation than previous measures for very large corpora. Evaluating on a lexical substitution task, we find that our contextualization method improves over a non-contextualized baseline across all parts of speech, and we show how the metaphor can be applied successfully to part-of-speech tagging. A number of ways to extend and improve the contextualization method within our framework are discussed. As opposed to comparable approaches, our framework defines a model of lexical expansions in context that can generate the expansions as opposed to ranking a given list, and thus does not require existing lexical-semantic resources.

Keywords: distributional semantics, lexical expansion, contextual similarity, lexical substitution, computational semantics

1 INTRODUCTION

In this article, we propose the new metaphor of two-dimensional text for data-driven semantic modeling of natural language and define a framework for its implementation. Being rooted in structural linguistics and distributional similarity, this metaphor provides a new angle on how to perform automated semantic processing. Whereas technically similar approaches have been pursued in the literature before, we feel that changing the viewpoint opens up new perspectives on how to advance the automated understanding of meaning in natural language.

The key element of this metaphor is the concept of *lexical expansion*. Lexical expansion generates additional lexical items for a given chunk of text, which enrich the textual representation and may be used in NLP (Natural Language Processing) tasks and applications. Expansion is performed for all present lexical items, and taking into account the textual context. Our approach constitutes a generative unsupervised model for semantic similarity in context that can be used to generate lexical expansions for unseen text material. These expansions help to bridge the lexical gap in semantics and serve as a valuable preprocessing step for many approaches in computational semantics, like word sense disambiguation, semantic text similarity, passage scoring and text segmentation.

After giving a short history of ideas that led from linguistic structuralism to the notion of distributional similarity and providing pointers to related work, we will map out the metaphor of two-dimensional text and explain the development from distributional to contextual similarity. Section 2 is concerned with operationalizing these notions in a scalable computational framework. In Section 3, we evaluate our methodology against a lexical resource and against a lexical substitution data set and show the value of the approach both for distributional as well as for contextual similarity. Sections 4 and 5 conclude and lay out possible points of departure for further work.

1.1 *From linguistic structuralism to distributional similarity*

What happens if we 'understand' language in the sense of assigning values of meaning to its elements, e.g. when reading a text? According to de Saussure (1916, 1959), our analysis happens from two distinct viewpoints: the *syntagmatic* viewpoint is concerned with assigning values based on the linear sequence of language elements, and the *associative (also: paradigmatic)* viewpoint assigns values according to the commonalities and differences to other language elements in the reader's memory.

> We see that the co-ordinations formed outside discourse differ strikingly from those formed inside discourse. Those formed outside discourse are not supported by linearity. Their seat is in the brain; they are a part of the inner storehouse that makes up the language of each speaker. They are associative relations. [...] The syntagmatic relation is in praesentia. It is based on two or more terms that occur in an effective series. Against this, the associative relation unites terms in absentia in a potential mnemonic series. (de Saussure, 1959, p.123)

In the metaphor of two-dimensional text, we propose to represent language in two dimensions: The first dimension is given by the linear nature of language, and represents syntagmatic relations between language elements, i.e. grammatical dependencies, positional relations or others. The second dimension contains language elements that are not present in the first dimension, but stand in paradigmatic relation to the language elements present. Figure 1 exemplifies possible associations for terms, and visualizes them in a second dimension, which we aim to model explicitly within our metaphor. The first dimension represents the linear sequence of language elements and their syntagmatic relations, the second dimension models associative relations that reside in the memory of the speaker/receiver. In this way, a text expansion step is realized.

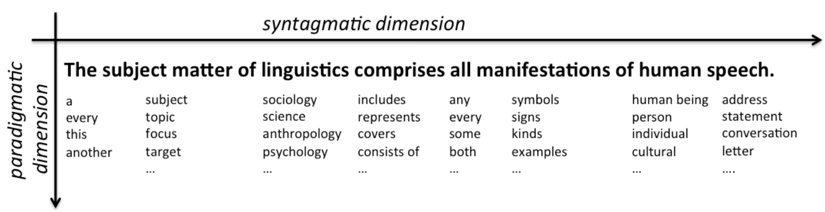


Figure 1: Exemplification of the metaphor of two-dimensional text

Please note that our metaphor specifies neither the language elements (words, terms, phrases etc.) nor the relation between the present elements and their expansions. The only constraint is that expansions in the paradigmatic relation share some commonality with their respective element. As de Saussure (1959, p.125) already states: “Mental association creates other groups besides those based on the comparing of terms that have something in common; through its grasp of the nature of the relations that bind the terms together, the mind creates as many associative series as there are diverse relations." From an application-based perspective in Natural Language Processing, it is easy to imagine that some of such relations might prove more useful than others when operationalizing the two-dimensional text for a given task. Further note that expansions in the paradigmatic dimension need to be contextualized to the present language elements. For example, in the sentence "almost all old subject case forms disappeared in French", "subject" would be expanded differently than is shown in Figure 1.

Many decades after the foundational work of Ferdinand de Saussure, Zellig S. Harris formulated his *distributional hypothesis*:

> The distribution of an element is the total of all environments in which it occurs, i.e. the sum of all the (different) positions (or occurrences) of an element relative to the occurrence of other elements. Two utterances or features will be said to be linguistically, descriptively, or distributionally equivalent if they are identical as to their linguistic elements and the distributional relations among these elements. (Harris, 1951, pp. 15f.)

Harris (1951) used the term *environments* to denote the language elements that stand in a syntagmatic relation to the element that is characterized. Note that an environment is not a language element, but an arbitrarily complex structure. However, we will approximate the environment with a tuple consisting of language elements and the syntagmatic relation, which we will call a *context feature*.

Whereas the distributional hypothesis was defined in the context of structural linguistics and originally formulated in order to identify phonetic variants of the same phoneme, it was not operationalized for computational semantics and cognitive science until about four decades later. After departing from an absolute notion of synonymy and instead focusing on semantic similarity as a graded notion, the *strong contextual hypothesis* of Miller and Charles (1991) states that "Two words are semantically similar to the extent that their contextual representations [context features] are similar". This suggests the following approach: using large text corpora to collect context features for language elements and comparing the extent to which these language elements share the same context features. This provides a way to compute semantic similarity without resorting to dictionary definitions or lexical resources. Miller and Charles (1991) were able to show that human judgments on semantic similarity as pioneered by Rubenstein and Goodenough (1965) correlate highly with the similarity of their context representations.

With the advent of large text corpora and reasonably precise methods to automatically assign grammatical structure to sentences, it became possible to compute term similarities for a large vocabulary (Ruge, 1992). Lin (1998) computed a *distributional thesaurus* (DT) by comparing context features defined over grammatical dependencies with an appropriate similarity measure for all reasonably frequent words in a large collection of text, and to evaluate these automatically computed word similarities against lexical resources. Entries in the DT consist of a ranked list of the globally most similar language elements (here: words) per language element of interest, which we call the *target*. While the similarities are dependent on the instantiation of the context feature as well as on the underlying text collection, they are global in the sense that the DT does not provide similarities with respect to particular occurrence of a target, but rather aggregates over all occurrences of the target and its similar elements.

We will build on the notion of the distributional thesaurus in our work, use the DT entries to populate the second dimension in the two-dimensional text representation, and move from the global notion of similarity to a contextualized version, which allows performing context-dependent text expansion for previously unseen target occurrences.

A similar review of the connection of de Saussurian linguistics and distributional similarity was presented in Sahlgren (2006). While Sahlgren motivated vector-space approaches to modeling meaning, we would like to stress that the two-dimensional text metaphor has not previously been employed as an approach to statistical semantics.

1.2 *Related work*

There has been a steady increase of interest towards incorporating distributional similarity into Natural Language Processing applications, particularly into language models. Whereas the workhorse of language modeling – the n-gram model – is a reliable and well-understood component in NLP systems, it models only very local properties of language and has been shown to be inadequate to grasp semantic dimensions of language such as ambiguity and synonymy (Biemann *et al.*, 2012).

Since local syntax could be modeled with a simple n-gram model, a desire to model semantics in a similarly straightforward fashion (i.e. trained from a background corpus without the need for linguistic theories, rule bases or knowledge bases) sparked a large body of research on semantic modeling. This includes computational models for topicality (Deerwester *et al.*, 1990; Hofmann, 1999; Blei *et al.*, 2003), and language models that incorporate topical (as well as syntactic) information (Boyd-Graber and Blei, 2008; Tan *et al.*, 2012). In the Computational Linguistics community, the vector space model (Schütze, 1993; Turney and Pantel, 2010; Baroni and Lenci, 2010) is the prevalent metaphor for representing word meaning. Vector space operations can be represented as vector and matrix operations, which makes this easily implementable due to the availability of tools such as MATLAB and libraries such as the GNU Scientific Library.

We do not agree that "nouns are vectors, and adjectives are matrices" (Baroni and Zamparelli, 2010), although they can of course be *represented* in these or similar ways. While vector space representations are becoming increasingly successful in modeling natural language semantics, vectors are typically too sparse and too highly dimensional to be used in their canonical form, and do not (naturally) encode relations beyond undifferentiated co-occurrence. We argue that there is no need to explicitly model non-existing relations, which would be zeros in the vector representation. We posit that it is only worthwhile storing properties for words or concepts if those same properties would be explicitly represented (non-zero) in a sparse representation.

Baroni and Lenci (2010) propose to store word-link-word triples in a tensor, and to produce vector spaces of various flavors by projection. While this model is a significant step towards a more generalized representation of (structured) vector spaces, it lacks the capability to address relations of higher complexity than single relations. Since in operationalizaton, similarity computations are carried out on pairs, we pursue a slightly different route in our holing system (see Section 2.1): we refrain from storing the tensor, and directly produce pairs from the observed structures in the text. Our formulation is thus able to produce the same behavior as the proposal of Baroni and Lenci (2010), but is more flexible and generic.

While computing semantic similarity on the basis of a background corpus produces a global model, which e.g. contains semantically similar words for different word senses of a target word, there are a number of works that aim at *contextualizing* the information held in the global model for *particular* occurrences. This is a similar task to word sense disambiguation against a lexical resource (Lesk, 1986), but without presupposing the existence of such a resource.

With his predication algorithm, Kintsch (2001) contextualizes the Latent Semantic Analysis (LSA) model (Deerwester *et al.*, 1990) for N-VP constructions by spreading activation over neighborhood graphs in the latent space. The Latent Dirichlet Allocation (LDA) model (Blei *et al.*, 2003) uses an inference step in order to adjust the topic distributions of the target occurrences. In particular, the question of operationalizing semantic compositionality in vector spaces (Mitchell and Lapata, 2008) received much attention and triggered shared evaluation tasks (Biemann and Giesbrecht, 2011; Padó and Peirsman, 2011): how can the (vector) representation of two lexical items be combined in context to yield an appropriate representation of their combination? Mixed results in favor of one or the other combination or mutual contextualization method (Mitchell and Lapata, 2008; Giesbrecht, 2009; Guevara, 2011) either indicate a dependency on the particular task, or raise questions regarding the representation itself.

Today's vector space representations suffer from two major shortcomings. First, size issues have to be handled with singular value decomposition (Golub and Kahan, 1965),[1] random indexing (Sahlgren, 2006) or other necessarily lossy dimensionality reduction techniques. Alternatively, efficient representations based on hashing functions (e.g. Goyal *et al.*, 2012) are employed to keep model estimation and computation at application time feasible. These issues arise as the word space is highly dimensional, and more structured variants (Padó and Lapata, 2007) that incorporate grammatical relations into the model lead to a further increase in the number of dimensions. Second, and more importantly, vector space models are not generative:

[1] The singular value decomposition is an algebraic factorization, which is used in LSA.

while impressive results are obtained when ranking a set of given alternatives by similarity of vector representation and context (e.g. word sense discrimination, Schütze 1998, synonyms, Rapp 2003, paraphrases, Erk and Padó 2008, word sense disambiguation, Thater *et al.* 2011), these tasks presuppose an existing list of alternatives to begin with.[2] Ideally, the alternatives should also originate from the model itself so as to avoid the manual creation of lexical resources for each language or application domain. We stress the need for a model that not only is able to rank given alternatives, but is also able to produce them.

2 OPERATIONALIZING SEMANTIC SIMILARITY

In this section, we describe how to operationalize semantic similarity. We describe a scalable and flexible computation of a Distributional Thesaurus (DT), and the contextualization of distributional similarity for specific occurrences of language elements (i.e. words). Care is taken to abstract away from particular preprocessing tasks needed for a given data set and from particular measures of similarity. Further, no assumptions regarding the size of the vocabulary nor the memory of the processors are made. For related works on the computation of distributional similarity, see Lin (1998), Gorman and Curran (2006), Lin and Dyer (2010), inter alia.

2.1 *Holing system*

To keep the framework flexible and abstract with respect to the preprocessing that identifies structure in language material (e.g. text or speech), we introduce the holing operation. Given a particular observation (structural representation) that has previously been extracted from the text (e.g. a dependency parse or an n-gram representation), the holing operation creates two distinct sets of observations: *language elements* (also referred as *terms*), and their respective *context features*. These two sets of observations form the basis for the computation of global similarities (Section 2.2) and for their contextualization (Section 2.3). Note that the holing operation is necessarily coupled to the

[2] Looping over the entire vocabulary to remove this restriction is neither computationally feasible nor plausible.

particular structural representation created by the pre-processing step, but all further steps towards contextual similarity abstract away from such pre-processing and operate on the same representation.

In the general case, an observation on the syntagmatic structure can be represented as an n-tuple containing an identifier of the observation, and the language elements that are part of the observation. We shall use the following sentence as the basis for examples:

```
Sentence:  I gave a book to the girl
Positions: 1 2    3 4    5  6   7
```

2.1.1 Observations

Let us now look at two different observations: dependency parses and token 4-grams. The collapsed dependency parse (Marneffe *et al.*, 2006) yields the following list of observations:

a) Dep.Parse:
(nsubj;$\mathtt{gave}_2$;$\mathtt{I}_1$), (det;$\mathtt{book}_4$;$\mathtt{a}_3$), (dobj;$\mathtt{gave}_2$;$\mathtt{book}_4$),
(det;$\mathtt{girl}_7$;$\mathtt{the}_6$), (prep_to;$\mathtt{gave}_2$;$\mathtt{girl}_7$)

Another pre-processing step that e.g. splits the language material into token 4-grams could produce these observations on the same sentence:

b) 4-gram:
($\$_0$;$\mathtt{I}_1$;$\mathtt{gave}_2$;$\mathtt{a}_3$), ($\mathtt{I}_1$;$\mathtt{gave}_2$;$\mathtt{a}_3$;$\mathtt{book}_4$),
($\mathtt{gave}_2$;$\mathtt{a}_3$;$\mathtt{book}_4$;$\mathtt{to}_5$), ($\mathtt{a}_3$;$\mathtt{book}_4$;$\mathtt{to}_5$;$\mathtt{the}_6$),
($\mathtt{book}_4$;$\mathtt{to}_5$;$\mathtt{the}_6$;$\mathtt{girl}_7$), ($\mathtt{to}_5$;$\mathtt{the}_6$;$\mathtt{girl}_7$;$\$_8$),
($\mathtt{the}_6$;$\mathtt{girl}_7$;$\$_8$;$\$_9$)

2.1.2 Holing operation

For a given set of observations extracted during pre-processing, a holing operation has to be defined that performs the split into language element(s) and context features. In the following examples the language element will be a word. However, the holing operation is not restricted to single words: arbitrary binary masks to define the parts of the observation tuples can be applied. For our example, we assume that we want to characterize single observed words a) by the dependency relation and the word it is connected to, and b) by the surrounding 4-gram context, where the observed word is located at the second position in the 4-gram. Further, we want to characterize pairs of observed words c) by their connecting two-edge dependency path. The application of the holing operation results in a set of pairs $< x, y >$ that identify the holing operation, as well as the parts it results in. The position of the language element x in its context tuple y is indicated by the hole symbol "@". For the single word examples, this could look like this:

a) Dep.Parse:
<I$_1$,(nsubj;gave$_2$;@)>, <gave$_2$,(nsubj;@;I$_1$)>, <book$_4$, (det;@;a$_3$)>,
<a$_3$, (det;book$_4$;@)>,..., <gave$_2$, (prep_to;@;girl$_7$)>,
<girl$_7$, (prep_to;gave$_2$;@)> .

b) 4-gram, second position:
<I$_1$, (\$$_0$;@,gave$_2$;a$_3$)>, <gave$_2$, (I$_1$;@;a$_3$;book$_4$)>,
<a$_3$, (gave$_2$;@;book$_4$;to$_5$)> , ..., <girl$_7$, (the$_6$;@;\$$_8$;\$$_9$)> .

For characterizing the pairs, the first part of the tuple is actually an ordered pair, and the second part contains two holes:

c) Dep.Parse two-edge paths:
<(I$_1$,book$_4$), (nsubj;gave$_2$;@$_1$;dobj;gave$_2$;@$_2$)>,
<(I$_1$,girl$_7$), (nsubj;gave$_2$;@$_1$;prep_to;gave$_2$;@$_2$)>,
<(gave$_2$,a$_3$),(dobj;@$_1$;book$_4$;det;book$_4$;@$_2$)>,
<(gave$_2$,the$_6$),(prep_to;@$_1$;girl$_7$;det;girl$_7$;@$_2$)>,
<(book$_4$,girl$_7$), (dobj;gave$_2$;@$_1$;prep_to;gave$_2$;@$_2$)> .

Note that a single observation can result in multiple pairs, as shown in a), where a dependency produces two pairs. Also, some observations need not produce any pairs, e.g. when deciding to exclude the `det` dependency relation, or constraining contexts along particular relations (cf. Lee, 1999).

The result of the holing operation, i.e. the list of pairs as shown above, is the only representation that further steps operate on. The pairs fully encode observed language elements and their contexts. For the computation of distributional similarity, the positional indices will be ignored, but they are required for the contextual expansion step.

The representation as shown here is more general than representations used by e.g. Lin (1998) and Curran (2004): whereas these previous works only allow a single term to be characterized with features, we allow arbitrary splits over arbitrarily complex observations, as shown in example c). This gives rise to the comparison of pairs, as e.g. conducted by Turney and Littman (2005) for extracting analogies of semantic relations in what they call *relational similarity*.

For the remainder of this paper, however, we mostly stick to the notion of *attributional similarity*, which is the basic element of the two-dimensional text expansion described above.

2.2 *MapReduce for similarity computation*

We now describe an implementation of the similarity computation for the Distributional Thesaurus (DT) based on the Apache Hadoop MapReduce framework,[3] which allows parallel processing of large (textual) data. The principle, developed by Dean and Ghemawat (2004), uses two steps, namely Map and Reduce. The Map step converts input text to key-value pairs, sorted by key. The Reduce step operates on all values that have the same key, producing again a data table with a key. As these steps do not require a global information flow, many Map and Reduce steps can be executed in parallel, allowing the system to scale to huge amounts of data. Further, we use Apache Pig,[4] a query language similar to SQL that allows us to perform database joins, sorting and limit operations on Hadoop data tables. To explain the workflow, we will refer to a holing system that extracts single terms as language elements for simplicity. However, the same workflow can be executed for more complex holing systems.

The data flow of the DT is illustrated in Figure 2. The example shown in this workflow uses a text file as input, where each line contains one sentence. The first MapReduce step in the workflow, called the *Context Feature Extractor*, implements a single holing operation as described in Section 2.1. For example, in Figure 2, the language element (which we will also call a term) is a word, concatenated with

[3] `http://hadoop.apache.org`

[4] `http://pig.apache.org/`

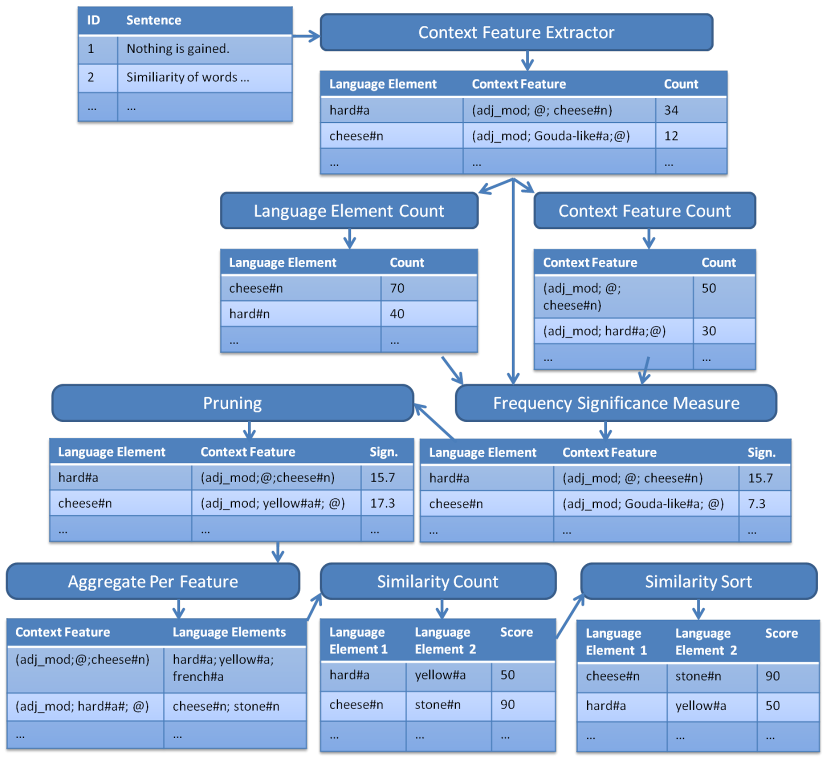


Figure 2: Workflow of the data processing using MapReduce

the corresponding part-of-speech; and the context feature is the dependency relation. Note that positional offsets are dropped here. For different holing operations (e.g. dependencies or 4-grams as in the previous section), the computation is executed separately.

In the next step, the frequencies of terms (*Language Element Count*) and single contexts (*Feature Count*) are collected, as they are needed to calculate the significance of each feature-term pair. For this work, we implemented different significance measures in *Frequency Significance Measure* and evaluate them in Section 3.2. For computing these measures, the tables produced by *Language Element Count* and *Feature Count* are joined to the table holding frequencies of term-feature pairs using an Apache Pig script. For a similar computation of word co-occurrences, Lin and Dyer (2010) propose to load the single frequencies into memory to avoid the join operation and to speed up the overall computation. While this works for a limited (albeit large) vocabulary of terms when carefully tuning the number of Mappers per computation node, this imposes a severe limitation on the number of (arbitrarily complex and productive) context features, which is why we do not adhere to this design pattern.

There are a total of three parameters for pruning the data during the *Pruning* step: t as a lower bound for the term-feature counts, s as a lower bound for the score of the respective significance measure, and p regulating the maximum number of context features per term. We argue that it is sufficient to keep only the p most salient features per term, as features of low saliency generally should not contribute much to the similarity of terms, and also could lead to spurious similarity scores. These pruning steps are especially important when using large data sets. The influence of the parameters on the quality of the DT will be examined in detail in Section 3.2.

Afterwards, all terms are aggregated by their features (*Aggregate Per Feature*), which allows us to compute similarity scores between all terms that share at least one feature (*Similiarity Count*). Here, we skip very frequent features (such as determiner modifiers), as they do not contribute meaningfully to similarities despite increasing computation time.

In comparison, Lin (1998) and Curran (2002) specify the similarity of terms using an "information" formula for each term-context relation and then calculate the similarity between terms using similarity measures. We show our similarity measure, as well as the measure used by Lin (1998) and a measure recommended by Curran (2002) in Table 1.

Function $f(.)$ returns the frequency of the selected element and $p(.)$ returns the probability. In contrast to the notation of Lin and Curran, we combine the relation name and the feature elements. To formulate Lin's information measurement in this notation, we define a $relation(.)$ function, which extracts only the relation name for a given context feature, and a $feature(.)$ function, returning all features for a term. Comparing our approach to other distributional similarity measurements (cf. Lee, 1999; Lin, 1998; Weeds, 2003), we do not need a "two-staged" formula, but can directly calculate the similarity by counting the overlap of features of two terms. This has the advantage that we do not need to calculate similarities between all pairs. Additionally, using only the p features per term having the

Table 1: Similarity measures used for calculating the distributional similarity between terms

Information measurements	
Lin's formula	$I(term,feature) = lin(term,feature) =$ $= \log \frac{f(term,feature)*f(relation(feature))}{\sum(f(word,relation(feature))f(word)}$
Curran's t-test	$I(term,feature) = ttest(term,feature) =$ $= \frac{p(term,feature)-p(feature)*p(term)}{\sqrt{p(feature)*p(term)}}$
Similarity measurements	
Lin's formula	$sim(t_1,t_2) = \frac{\sum_{f \in features(t_1) \cap features(t_2)} (I(t_1,f)+I(t_2,f))}{\sum_{f \in features(t_1)} I(t_1,f) + \sum_{f \in features(w_2)} I(w_2,f)}$
Curran's dice	$sim(t_1,t_2) = \frac{\sum_{f \in features(t_1) \cap features(t_2)} \min(I(t_1,f),I(t_2,f))}{\sum_{f \in features(t_1) \cap features(t_2)} (I(t_1,f)+I(t_2,f))}$
Our measure	$sim(t_1,t_2) = \sum_{f \in features(t_1) \cap features(t_2)} 1$
w. filtering	$sim(t_1,t_2) = \sum_{\substack{f \in rankedfeatures(t_1,p) \cap rankedfeatures(t_2,p) \\ f(t_1)>t \wedge f(t_2)>t \\ score(f)>s \wedge score(f)>s}} 1$

highest significance scores (which are retrieved using the function *rankedfeatures*$(term, p)$) speeds up our approach tremendously and acts as a noise filter.

This constraint makes this approach more scalable to larger data, as we do not need to know the full list of features for a term pair at any time. As we will demonstrate in Section 3, this simplification does not impair the quality of the obtained similarities, especially for very large corpora.

The last step sorts the list by term and by descending score. To reduce the size of the output, only the most similar n terms per entry are kept. The overall computation results in second order (paradigmatic) similarity scores that are ready to be imported to a storage database, as to be accessible for the contextualization component. Further, we store the first order (syntagmatic) significant pairs $< x, y >$, together with their significance score, as we will need them for contextualization.

Our small Hadoop cluster (64 cores on 8 servers) was able to perform the entire computation (excluding pre-processing, i.e. parsing) of our similarity measure for the whole vocabulary of our largest corpus of 120 million sentences in well under a day. Within our framework, we also provide Pig scripts for the computation of other similarity measures (cf. Table. 1), although they take much longer to compute. The implementation is available via the JoBimText[5] project as open-source software under the ASL 2.0 for download.

2.3 *Contextualizing distributional similarity*

Now, we explore a way of contextualizing semantic similarity. The task of contextualization is cast as a ranking problem (in accordance with most literature on lexical substitution): given a set of candidate expansions as provided by the DT, we aim at ranking them so that the most similar terms in context will be ranked higher. Intuitively, candidates that are not compatible with the given context should be ranked lower, whereas candidates that fit well should land on top of the list.

When expanding a target, we run the holing system on the lexical material containing our target, and select all pairs $< x, y >$ where $x =$ target. Further, we obtain a set of candidate expansions X' by selecting the most similar n terms from the DT entry of the target. For each pair, we iterate over the elements x' in X' and retrieve the significance score of $< x', y >$. If the candidate expansion has been observed in the context of y before, this will result in a positive score. If the candidate has not been observed, it is probably incompatible with y and gets assigned a score of 0 for this context. In this way, each candidate x' gets as many scores as there are pairs containing x in the holing system output. An overall score per x' is then calculated as the harmonic mean of the add-one-smoothed single scores. Smoothing is necessary to be able to rank candidates x' that are not compatible with all contexts.

In Figure 3, we illustrate this using the noun target "cold" in the sentence "I caught a nasty cold". Our dependency-parse-based holing system produced the following pairs for "cold":

<cold$_5$,(amod;@;nasty$_4$)>, <cold$_5$,(dobj;caught$_2$;@)> .

The top 10 candidates for "cold" as a noun are $X' = \{$heat, weather, temperature, rain, flue, wind, chill, disease$\}$. In Figure 3, the scores per pair are listed: e.g. the pair `<heat,(dobj;caught;@)>`

[5] `http://sourceforge.net/p/jobimtext/wiki/Home/`

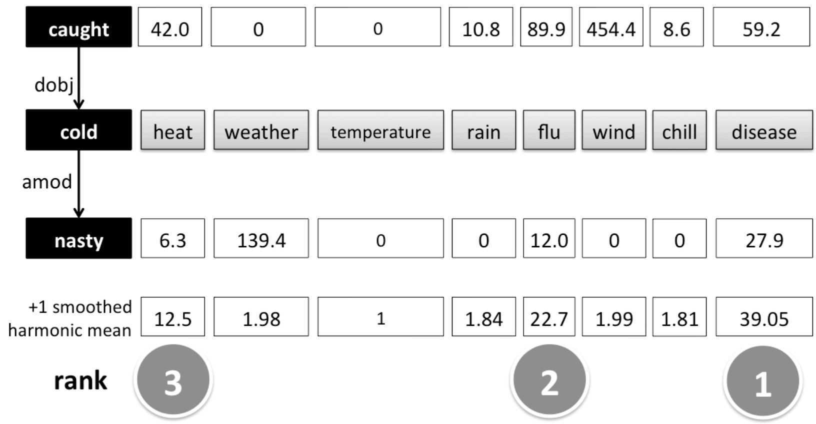


Figure 3: Contextualized ranking for target “cold” in the sentence “I caught a nasty cold” for the 10 most similar terms from the DT (here: 10 million sentences, LMI, $p = 1000$)

has a Lexicographer’s Mutual Information (LMI) score of 42.0, the pair <`weather,(amod;@;nasty)`> has a score of 139.4, and the pair <`weather,(dobj;caught;@)`> was not contained in our first-order data. Ranking the candidates by their overall scores as given in the figure, the top three contextualized expansions are “disease, flu, heat”, which are compatible with both pairs. For $n = 200$, the ranking of fully compatible candidates is: “virus, disease, infection, flu, problem, cough, heat, water”, which is clearly preferring the disease-related sense of “cold” over the temperature-related sense.

Context features differ in their usefulness: a context feature like `(det; @; a)` is much less useful for ranking expansions than context features with more specific language elements, such as `(amod; tasty; @)`, which e.g. selects edibles and thus could distinguish between “Turkey” the country and “turkey” the bird. To compensate for this effect, we found it advantageous to divide the score by the corpus frequency of the context feature language element, and to only take context features containing content words (i.e. nouns, verbs, adjectives) into account. Of course, many more weighting schemes would be possible.

Iterating the per-word expansion over the whole sentence to expand all the terms yields a two-dimensional contextualized text.

3 EVALUATING TWO-DIMENSIONAL TEXT

Directly evaluating the quality of a (non-contextualized) DT is intrinsically hard. It is known that distributional similarity somewhat reflects semantic relations in lexical resources, but it is clear that a DT will never correspond exactly to a lexical resource, e.g. for the reasons of vocabulary mismatch, skewed word sense distributions in the underlying collection and rare senses in the resource, cf. Curran (2002) and Henestroza Anguiano and Denis (2011). We follow a pragmatic approach and evaluate DTs of different parameterizations against WordNet, using a new path-based approach. While the aforementioned shortcomings make it hard to draw conclusions about the absolute quality of the DTs, our evaluation methodology still allows to compare DTs relatively to each other.

Regarding the contextualization, we chose to evaluate our technique in lexical substitution tasks. We stress again that – as compared to previous methods – we do not use a lexical resource for substitution candidates, but generate them using the DT. Therefore, our overall system solves a harder task than merely ranking a given set of alternatives.

Finally, we show how to apply our two-dimensional text processing to an existing NLP system that performs part-of-speech tagging in Section 3.4. In the same way, other existing NLP components could be extended by this two-dimensional representation.

3.1 *Data sets and methodology*

For DT evaluation, we use a word list of English nouns of varying frequency. For evaluation of the contextualization, we use two different lexical substitution data sets. We briefly describe the two datasets and the metrics we used in each case:

- **1000 frequent and 1000 infrequent nouns using WordNet path similarity**
  To evaluate our method under several parameter settings and against previous measures, we use the list of 1000 frequent and 1000 infrequent nouns from the British National Corpus previously employed in Weeds (2003). To calculate similarity scores between these target words and the most similar words in the distributional thesauri, we use the WordNet::Similarity path measure

(Pedersen *et al.*, 2004). For pairs of words that are members of several synsets, we use the shortest path between them. While the path measure has been criticized because of the varying granularity in different regions of WordNet, it is well-suited for relative comparison and has an intuitive interpretation: two words are fairly similar if the shortest route between them is small, and are less similar if the shortest route between them is long.

- **Lexical Substitution Task 2007 dataset (LexSub)**
  The LexSub[6] data were introduced in the Lexical Substitution task at Semeval 2007 (McCarthy and Navigli, 2009). It consists of 2010 sentences for a total of 201 target words (10 sentences for each word). For each target in context, five English native speaker annotators were asked to provide as many paraphrases or substitutions as they found appropriate. This way, valid substitutions are assigned a weight (or frequency) which denotes how many annotators suggested that particular word. We used the evaluation methodology as provided by the task organizers, tuned our approach on the trial data (300 sentences), and evaluated on the official test data (1710 sentences).

3.2 *Distributional similarity*

For computing the DT, we used newspaper corpora of up to 120 million sentences (about two gigawords), compiled from freely available corpora from LCC[7] and from the Gigaword corpus (Parker *et al.*, 2011). We examine the influence of the corpus size by computing DTs on corpora of different magnitudes, and evaluate the influence of parameters and significance measures.

3.2.1 Evaluation methodology

In this work, two different holing systems were used in the first step of the DT computation:

- As a simple baseline holing system, we employ token bigrams: for each token, the preceding and the following word are used as con-

[6] `http://nlp.cs.swarthmore.edu/semeval/tasks/task10/data.shtml`

[7] Leipzig Corpora Collection, `http://corpora.uni-leipzig.de`, (Richter *et al.*, 2006).

text features. This holing system uses information that is equivalent to the information available in a bigram language model.

- As a more informed holing system, we use collapsed dependency parses from the Stanford parser,[8] as depicted in Figure 2 and as described in Section 2.1.

To avoid confusion between words with different part-of-speech (POS) tags, we do not use the word itself, but rather the lemmatized[9] word combined with a POS tag[10] for both holing systems.

For all corpora, we only calculated similarities based on single word expressions and did not address multiword expressions, which is subject to further work. For this reason, we ignored multi-word entries in our evaluation data sets entirely.

3.2.2 Evaluation of DT parameters

In an initial exploration, we use 10 million sentences from the LCC to compute DTs for different parameters. We do not filter on occurrence frequency t and significance thresholds s, but merely vary the number of context features per term p. This parameter has a direct consequence for the run-time of the DT computation and the intermediate and final disk space.

To rank context features by their significance, we compare three significance measures,[11] two of which we show in Table 2:

- PMI Pointwise Mutual Information: a widely used significance measure since its introduction to NLP by Church and Hanks (1990).
- LMI Lexicographer's Mutual Information (Kilgarriff *et al.*, 2004), also known as Local Mutual Information (Evert, 2005): since PMI is known to assign high significance scores to pairs formed by low-frequent items, the LMI measure tries to balance this by multiplying the PMI score with the pair frequency.

[8] `http://nlp.stanford.edu/software/lex-parser.shtml`, (Marneffe *et al.*, 2006).

[9] The verbs, nouns and adjectives are lemmatized, using a Compact Patricia Trie classifier (Biemann *et al.*, 2008) trained on the verbs, nouns and adjectives.

[10] As produced by the Stanford parser.

[11] For a comparison of measures, see e.g. Evert (2005) and Bordag (2008).

- LL Log-likelihood: also a widely used measure since it was introduced by Dunning (1993), known to be less susceptible to overestimation of low frequency pairs. We omit its lengthy expanded formula here, which can be found e.g. in Bordag (2008).

Table 2: Significance measures used to rank the term feature pairs

PMI	$\mathrm{PMI}(term,feature) = \log_2\left(\frac{f(term,feature)}{f(term)f(feature)}\right)$
LMI	$\mathrm{LMI}(term,feature) = f(term,feature)\log_2\left(\frac{f(term,feature)}{f(term)f(feature)}\right)$

The results are calculated based on the 1000 frequent and 1000 infrequent target nouns. Average WordNet path similarities are computed between the target and the highest-ranked 5 and 10 words in its DT entry that occur in WordNet. For words invoking several synsets, we compute all possible pairs and use the minimal path distance. The results for the 1000 frequent nouns are shown in Table 3.

Note that the PMI measure does not play well with our pruning scheme regulated by the p parameter: while the other two measures yield very similar scores, PMI produces clearly inferior results. This confirms previous observations that PMI overestimates context features with low frequency: these context features might characterize the terms extremely well, but are too sparse to serve as a basis for the computation of second-order similarity (cf. Bordag, 2008). For high-frequency words, the most significant context features ranked by PMI are largely rare contexts of high specificity, whereas for low-frequency

Table 3: Wordnet Path Similarity for 1000 frequent nouns for DTs computed on 10 million sentences

Top	Sign.	max number of context features p				
words	Meas.	10	100	300	500	1000
top10	LL	0.04178	0.25744	0.27699	0.27635	0.27574
top10	LMI	0.03636	0.25449	0.27746	0.27554	0.27530
top10	PMI	0.00000	0.00213	0.04480	0.09104	0.16877
top5	LL	0.12034	0.29345	0.31106	0.31515	0.31182
top5	LMI	0.11666	0.29272	0.31378	0.31307	0.31028
top5	PMI	0.00000	0.00510	0.05836	0.11063	0.19268

words, this problem is less severe since there are fewer contexts to begin with, and so the top 1000 PMI contexts contain enough context features to produce similarities almost on par with the other measures.

More interestingly, there seems to be an optimal value for p, as more context features apparently do not improve the similarity and the highest values are obtained for $p = 300$ in this experiment. However, degradation for larger values of p is small. Values for average path similarities over the top 5 words are consistently higher than for the top 10 words, indicating that the ranking is valid with respect to semantic closeness.

Looking at the results of the infrequent nouns (see Table 4), we observe much lower average values throughout.

This is partially due to the words in the given noun list that do not have an entry in the DT at all; but more plausibly the lack of overall data for these words causes less reliable similarities. A further reason is the incomplete WordNet coverage for senses that are dominant in our collection. For example, the word *anime* belongs to two synsets: "a hard copal derived from an African tree" and "any of various resins or oleoresins", whereas an entry for *anime* in the sense of the Japanese animation movie is missing. The entries of the DT using LMI and $p = 500$ contains "novel, music, manga, comic, cartoon, book, film, shows, sci-fi", which all receive a low score. For infrequent words, the difference between PMI and the other measures is much less pronounced, yet we can still safely conclude from these experiments that PMI is not the optimal measure in our setup.

Table 4: Wordnet Path Similarity for 1000 infrequent nouns for DTs computed on 10 million sentences

Top	Sign.	max number of context features p				
words	Meas.	10	100	300	500	1000
top10	LL	0.03252	0.18560	0.20426	0.20572	0.20238
top10	LMI	0.03349	0.18516	0.20315	0.20577	0.20373
top10	PMI	0.00000	0.05892	0.14757	0.16597	0.16931
top5	LL	0.09268	0.21497	0.23231	0.23680	0.23108
top5	LMI	0.09469	0.21512	0.23208	0.23541	0.23179
top5	PMI	0.00012	0.10502	0.17446	0.18966	0.19318

For the next experiment, we examine the influence of corpus size and the difference between using dependency parses or neighboring tokens, again evaluating against our set of frequent and infrequent nouns using WordNet path similarity. Figure 4 displays the average WordNet path similarity score for the top-ranked five words for the 1000 frequent nouns (infrequent nouns show qualitatively similar results).

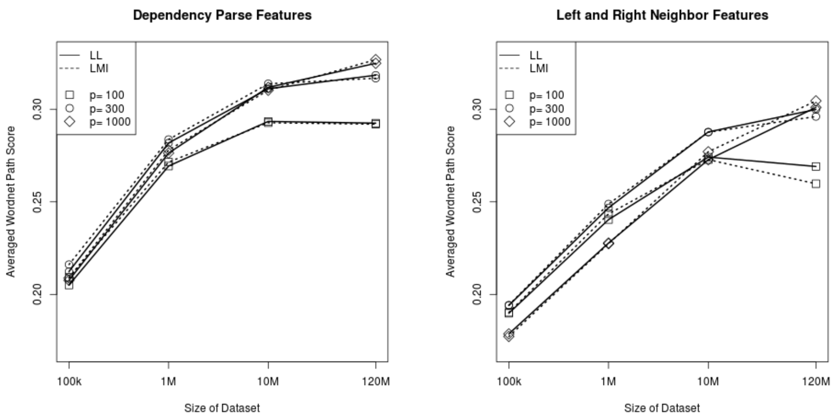


Figure 4: Corpus size vs. WordNet path similarity for different max. numbers of context features p, comparing LMI and LL measures, for two holing systems

As a general trend, larger corpora call for larger p – an effect that is especially pronounced for the token bigrams: whereas $p = 100$ produces the best results on the 1M sentence corpus, $p = 300$ excels for 10M sentences and the best scores overall for 120M sentences are obtained with $p = 1000$. However, differences between $p = 500$ and $p = 300$ respectively $p = 1000$ are small, so choosing p in the range of 500–1000 can be recommended for very large corpora. Comparing the holing systems, the dependency parse features result in much higher performance for small corpora, but do not outperform bigram features on large corpora by a great extent. This is consistent with a previous, similar evaluation by Curran (2004).

To support our qualitative observations, we list the DT entries for the LL measure and $p = 1000$ for the frequent noun "answer" and for the rather infrequent noun "tint" for different corpus sizes in Figures 5

Target: **answer**

r	100K	WP	1M	WP	10M	WP	120M	WP
1	question	1/5	solution	1	solution	1	explanation	1/3
2	reason	1/4	outcome	1/7	response	1	response	1
3	solution	1	explanation	1/3	explanation	1/3	reply	1
4	guy	1/11	way	1/6	question	1/5	solution	1
5	deal	1/4	excuse	1/6	reply	1	conclusion	1/4
6	decision	1/7	reaction	1/4	information	1/4	description	1/3
7	money	1/10	response	1	thing	1/3	question	1/5
8	plan	1/10	copy	1/6	rationale	1/12	information	1/4
9	story	1/4	thing	1/3	choice	1/6	remedy	1/10
10	goal	1/9	truth	1/3	reason	1/4	retort	1/3
Ø		0.25		0.41		0.46		0.48

Figure 5: DT entries for "answer" with WordNet path similarities (WP), comparing different corpus sizes from 100K sentences up to 120M sentences

Target: **tint**

rank	100K	WP	1M	WP	10M	WP	120M	WP
1	–	–	button	1/12	color	1/2	hue	1/5
2	–	–	clothing	1/13	hue	1/5	shade	1
3	–	–	meat	1/10	tone	1	color	1
4	–	–	suit	1/12	shade	1	tinge	1/2
5	–	–	arrow	1/12	tinge	1/2	shading	1/14
6	–	–	beer	1/16	hair	1/10	texture	1/4
7	–	–	berry	1/14	glow	1/7	tone	1
8	–	–	blazer	1/18	haze	1/11	coloration	1/3
9	–	–	box	1/10	light	1/4	palette	1/8
10	–	–	carpet	1/12	odor	1/5	patina	1/14
Ø		0		0.08		0.40		0.41

Figure 6: DT entries for "tint" with WordNet path similarities (WP), comparing different corpus sizes from 100K sentences up to 120M sentences

and 6. We provide the WordNet path similarities in fractional notation, where $1/x$ indicates a path length of $x-1$ between target and similar term.

It is apparent that for a frequent word like "answer", already a small collection can produce some reasonable top-ranked words, yet the list quickly degrades for 100K and 1M sentences. A typical effect for the largest of our corpora is illustrated with "retort", which is about 20 times less frequent than "answer", yet can collect enough significant contexts to enter its top 10 list. We frequently observed rather rare hyponyms and co-hyponyms of targets in the DTs computed from 120M sentences, which tremendously increases coverage for applications.

Looking at another example, the noun "tint" is too infrequent to receive any entry in the 100K sentence DT, and has a rather random collection of words for 1M sentences, stemming from the shared adjective modifier "dark". The larger collections produce quite suitable lists, again with a higher specialization for the 120M sentence corpus.

Next, we compare our similarity measure to similarities based on Lin's and Curran's measures, as introduced in Section 2.2. For both LL and LMI, we fixed $p = 1000$.

According to the results shown in Table 5, we can see that our method leads to much better results for frequent words.

In the evaluation of the 100k sentence dataset we observe that Lin's measure beats all other measures for the frequent words. For this small corpus, our measure is the second best measure and Curran's measure leads to the lowest scores. For infrequent nouns, our approach produces the best results for this dataset. For the 120M sentence dataset, Lin's measure and our measure produce similar results, with our method being at slight advantage. Curran's measure shows inferior performance. We can observe that all measures improve when based on larger data. It seems surprising that our comparably simple measure matches and outperforms, respectively, two well-established measures from the literature. We will spend the remainder of this section discussing possible reasons.

Since Lin's measure was optimized on a much smaller corpus of about three million sentences using a different parser in Lin (1998),

Table 5: Wordnet Path Similarity for 1000 frequent and 1000 infrequent nouns, computed on 100K and 120M sentences comparing our measure to measures by Lin (1998) and Curran (2002)

corpus	Freq./	Top	Other methods		Our method	
size	infreq.	words	Lin	Curran	LL	LMI
100k	freq	top 10	**0.21322**	0.17779	0.19566	0.19645
100k	freq	top 5	**0.23295**	0.18031	0.20736	0.20798
100k	infreq	top 10	0.08186	0.09565	**0.12239**	0.12213
100k	infreq	top 5	0.10128	0.10164	**0.12759**	0.12683
120M	freq	top 10	0.27874	0.25429	0.28270	**0.28339**
120M	freq	top 5	0.31742	0.28355	0.32479	**0.32679**
120M	infreq	top 10	0.21480	0.17829	**0.22139**	0.21902
120M	infreq	top 5	0.24640	0.19490	0.25773	**0.25798**

it seems to be reasonable to assume that the factor regarding the frequency of the relation $f(relation(feature))$ (cf. Table 1) suppresses the influence of noise, but at the same time puts too much emphasis on frequent relations, which prevents a more fine-grained characterization of items by features. This is also confirmed by the results based on the 100k dataset. Our measure, on the other hand, increases in quality when more evidence (higher frequency) is available, which results in higher quality overall as collections are scaled up, and the p parameter on the number of characterizing features takes care of the noise.

Curran's measure was optimized on a collection larger than that in Lin's work, measuring about 300 million words (15 million sentences, Curran 2002), which is still about one order of magnitude smaller than our large corpus. Surprisingly, we could not confirm that Curran's measure performs better than Lin's measure (Curran, 2002).[12] This might be explained by the use of a different parser and different test words. Additionally, Curran uses a different evaluation method, as he compares his DT against entries from a combined set of entries taken from various thesauri, and only using a small number of nouns.

Wrapping up the DT evaluation, we can state that the most important factor for obtaining a high-quality DT is the amount of data. Comparing our proposal with existing measures, we feel that the effectiveness of semantic similarity measures on large corpora has been reconfirmed: on more data, simpler measures perform as well or even better than measures that were intended to give good results for small collections – an insight similar to that described in the seminal work of Banko and Brill (2001) for machine learning methods.

When using our measure, which is highly optimized for speed of computation, a suitable significance measure for ranking context features is required: measures that favor frequent items are preferable in our setup. Here, LMI and LL produced very similar scores, hence LMI is preferable because of its simpler, and thus more efficient, computation. There is no need to retain more than 500–1000 context features

[12] Following his Dice formula, it is not clear whether to take the intersection or the union of the features of two words. We tested different possibilities that, however, did not yield improvements. We decided to use the intersection, as it is unclear how to interpret the minimum function otherwise.

per term even for large corpora, which allows us to speed up the computation of the DT by a large degree. Equipped with this result, we can proceed to evaluate the effects of contextualization.

3.3 *Contextual similarity*

The contextualization evaluation was performed using the distributional thesaurus that was compiled using up to 120M sentences and using the LMI measure and $p = 1000$, as this combination showed the best performance in the previous section. The outcome for the contextualization is shown using the test set of the LexSub dataset, described in Section 3.1.

3.3.1 Evaluation methodology

For the evaluation of the LexSub dataset we used the out of ten (OOT) precision and OOT mode precision on the LexSub test set of 1710 sentences, as described in McCarthy and Navigli (2009). The OOT measure allows us to make up to 10 guesses, discarding further guesses. Both measures calculate how many substitutions have been detected within ten guesses over the complete subset. The difference is the “detection” of a correct match per entry. Whereas the OOT precision sums up the number of correct guesses divided by the number of possible answers, in the OOT mode precision evaluation the system is credited if the mode from the annotators (most frequent response(s)) is found within the system’s 10 responses. We do not apply any special handling regarding multiwords (terms consisting of more than one word), which are not contained in our DT and are therefore always missed. For comparison, we use the results of the distributional thesaurus as a baseline to evaluate the contextualization. Note that our system does not yield duplicate entries, which are known to influence the OOT metric. We chose the OOT measure over the ‘best’ metric, since it better fits the metaphor of expanding text with several words.

As already mentioned in Section 2.3, we only use context features that contain another content word[13] and divide the weight by their corpus frequency. Furthermore, we use a threshold for the significance value of the LMI values of 40.0, and the most similar 30 terms from the

[13] Words with part-of-speech prefixes V, N, J, R.

DT entries as candidates for the contextual ranking. These parameters have been determined by optimizing OOT scores on the LexSub trial set.

3.3.2 Results

Since it can be expected that the contextualization algorithm is dependent on the number of context features for the target occurrence, we report scores for targets with at least two and at least three dependencies separately. In the LexSub test data, all targets have at least one, 49.2% of the targets have at least two and 26.0% have at least three dependencies. Furthermore, we also evaluated the results broken down into separate parts-of-speech of the target. The results for the OOT precision and the mode precision for both the entries of the distributional thesaurus (DT) and the contextualization (CT) are shown in Table 6.

Table 6: Results on the LexSub test dataset for global (DT) and contextualized (CT) similarities, per min number of dependencies to target

		Precision			Mode precision		
min. # dep.		1	2	3	1	2	3
POS	Alg.						
adjective	DT	32.81	33.64	35.02	43.56	43.53	42.86
adjective	CT	33.27	35.41	36.08	44.48	48.24	46.43
noun	DT	25.29	25.00	28.07	35.06	34.48	36.76
noun	CT	26.76	26.67	28.63	39.08	38.92	39.71
verb	DT	24.41	22.63	22.10	30.00	29.35	29.14
verb	CT	24.48	24.33	23.80	32.58	33.33	34.29
adverb	DT	28.85	26.75	29.88	41.43	34.38	66.67
adverb	CT	20.80	29.46	36.23	30.48	40.63	100.00
ALL	DT	**27.48**	25.10	25.72	37.19	33.39	33.77
ALL	CT	27.02	**26.84**	**27.14**	**37.35**	**37.75**	**38.41**

Inspecting the results for precision and mode precision without filtering entries regarding parts-of-speech (denoted as ALL), only marginal changes can be seen for entries having at least one dependency. But we observe substantial improvements for targets with more than one dependency: more than 1.6 points in precision and more than 4 points in mode precision.

The results regarding different part-of-speech tags of the target words follow a similar trend. For adjectives, nouns and verbs, the contextualization improves results throughout for all targets. Most notably, the largest relative improvements are observed for verbs, which is a notoriously difficult word class in computational semantics. For adverbs, contextualization hurts in cases where the adverb has fewer than two context features, but helps for targets with a minimum of two dependencies. Since there are merely seven instances where adverbs have at least three dependencies in the dataset, the high scores in mode precision are probably not representative.

Regarding performance on the original lexical substitution task (McCarthy and Navigli, 2009), we did not come close to the performance of the participating systems, which range between 32–50 precision points and 43–66 mode precision points (only taking into account systems without duplicate words in the result set). However, all participants used one or several lexical resources for generating substitution candidates, as well as a large number of features. Our system, on the other hand, merely requires a holing system – in this case based on a dependency parser – and a large amount of unlabeled text, as well as a very small number of contextual clues. Scores for a DT computed on the British National Corpus using Lin's measure as reported in McCarthy and Navigli (2009) are slightly higher than what we observe here, which we attribute to a different underlying background corpus.

3.4 *Two-dimensional representation for part-of-speech tagging*

In this section, we demonstrate how the notion of two-dimensional text can be used directly in NLP tasks using part-of-speech (POS) tagging as an example. While POS tagging is generally regarded as solved for languages and domains with sufficient amounts of training data, there are still challenges in domain adaptation, e.g. for user-generated content (Gimpel *et al.*, 2011) or for domain-specific texts (e.g. Biemann 2009 reports a 20% out-of-vocabulary (OOV) rate of news models on medical texts). The largest source of errors in POS assignment is observed for out-of-vocabulary words, i.e. words that were not contained in the training data and have to be classified according to context and surface features only. A sequence of OOV words can throw off the sequence classification algorithm, resulting in poor performance. For classifiers that do not normalize over the whole sequence, this has been described as the label bias problem, cf. Lafferty *et al.* (2001).

Two-dimensional text provides a possibility to overcome the OOV problem by resorting to the most similar in-vocabulary word, when encountering a word unseen in training. For this, merely a list of in-vocabulary words has to be maintained. Presupposing an existing supervised POS tagger, the scheme is executed as follows.

Model training

1. Train the POS tagger on training text and construct the list of in-vocabulary words.
2. Compute a distributional thesaurus (DT) on a large background corpus.

POS tagging task

1. Determine the OOV words of the input text by checking the in-vocabulary word list.
2. For all OOV words, replace the OOV word by its most similar in-vocabulary word according to the DT.
3. Tag the altered text with the POS tagger, and project tags back to the original text.

For our experiments, we trained the well-known TreeTagger (Schmid, 1995) on the Penn Treebank (PTB, Marcus *et al.* 1993), following Collins (2002) by training on Sections 0–18 and testing on Sections 22–24.[14] The distributional thesaurus was induced on 120M sentences of English newswire, using a holing system based on word trigrams: the center word of each trigram served as the word, the two neighboring words (left and right together) served as the context. We retained the most similar 100 words per entry.

Figure 7 illustrates this method using an example: in the sentence "Renting out an unfurnished one-bedroom triplex in San Francisco", the words "unfurnished", "one-bedroom" and "triplex" are OOV words, not being part of the PTB training set. In the case of "one-bedroom" this might seem surprising, but the Penn Treebank consistently uses a spelling without the hyphen, resulting in two tokens

[14] We do not perform parameter optimization and therefore do not use Sections 19–21, which are normally used for development.

Figure 7: Illustrating the two-dimensional extension for POS tagging

"one bedroom". While the top-most similar words to "unfurnished" and "triplex" ("empty" and "duplex") are in-vocabulary words of our POS tagger, the most similar in-vocabulary word for "one-bedroom", "two-room", is the third most similar expansion according to our DT. Tagging the alternate sentence "Renting out an empty two-room duplex in San Francisco" results in correct assignment of POS tags, cf. Figure 7.

Evaluating the improvement over the whole test set, we improved the accuracy on the 3562 OOV words (the majority of them are verbs, nouns and adjectives) from 37.82% to 74.12%.[15] Overall, the accuracy of the tagger improved from 95.28% to 96.07%, only by altering the tagging strategy on the portion of 2.1% OOV words.

This overall performance is well below state-of-the-art POS tagging on this dataset (which is at 98.5%, Søgaard 2011), where successful approaches make heavy use of surface feature backoff, word clustering on background corpora, and advanced machine learning techniques. Our setup, however, illustrates how the metaphor of two-dimensional text can be used in the context of existing NLP software, while neither needing to alter the feature representation nor the learning algorithm for machine learning. The key, and its novelty with respect to word-space approaches, is that the DT is able to *generate* the most similar words, so that they can be used in lieu of words that impose difficulties for the software (i.e. OOV words for POS tagging). A comparable approach of expanding text representations with similar words from our process was successfully used by Miller *et al.* (2012) for state-of-the-art knowledge-based all-words word sense disambiguation.

---

[15] We enabled the heuristics of the TreeTagger (-hypen-heuristics, -ignore-prefix, -cap-heuristics) which improved the accuracy by 0.15% without any OOV replacement.

4 FUTURE WORK

There are a number of ways in which our framework for the metaphor of two-dimensional text can be filled and extended. In the remainder, we will briefly describe approaches that we intend to try in the future.

4.1 *Generalization of the holing system*

Experiments presented here used holing systems that extract context features for single words. While it is straightforward to extend it to pre-defined multi-word units, it would be promising to allow arbitrary, not necessarily contiguous sets of language elements, and determine their appropriateness by means of the similarity computation. The current framework also supports the computation of context feature similarities by exchanging the columns "language elements" and "context features" in the DT computation depicted in Figure 2, yet it still needs to be worked out how similarities of contexts could be used in the contextualization. Along these lines, a further generalization of the holing system is to use an arbitrary number of holes, which could e.g. allow us to detect similarities between active and passive constructions.

4.2 *Combination of signals for contextualization*

While we have only shown experiments using a single holing system at a time, it is possible to combine signals from several holing systems for contextualization, as well as signals from other semantic models such as topic models (cf. Thater *et al.*, 2011). Further, there is a large space of parameterization of the holing system with respect to the use of priors, the numerical transformation of word-context-significances to path probabilities, and the weighting of signals from different models.

4.3 *Other sampling methods for contextualization*

While we have demonstrated that a simple contextualization method as described in Section 2.3 is already able to achieve improvements of the lexical expansion quality, we would like to employ sampling methods that adjust path probabilities based on previous trials, like Metropolis-Hastings Sampling (Hastings, 1970), or dynamic programming approaches to compute the ranking of expansions efficiently (cf.

Viterbi, 1967; Lafferty *et al.*, 2001). In contrast to our simple method, these approaches normalize over the whole expanded sequence and perform expansions for all terms at the same time.

4.4 *Word sense induction clustering*

As the contextualization was described, the problem of word sense disambiguation is handled implicitly by down-ranking lexical expansions that refer to the wrong sense of the word in the context. It might be advantageous, however, to add word sense induction clustering on the DT entry (cf. Schütze, 1998; Widdows and Dorow, 2002; Biemann, 2010), and to perform the contextualization per cluster instead of per word to alleviate sparsity. Note that this per-entry clustering is different than the whole-vocabulary clustering proposed by Pereira *et al.* (1993) and others.

4.5 *Distinguishing expansions by patterns*

While word sense induction can distinguish similar words in the DT by sense, we need something else in order to obtain typed relations between a target and its potential expansions. One way of typing is to examine what patterns (e.g. is-a, part-of, antonyms) are common between target and expansion in our large corpus. These types would be useful for targeting certain types of expansions, e.g. excluding antonyms for lexical substitution. To keep the approach unsupervised and knowledge-free, we would like to find the patterns automatically in a co-clustering approach based on terms and patterns (Dhillon, 2001) rather than using pre-defined patterns (Hearst, 1992; Lin *et al.*, 2003).

4.6 *Machine learning on delexicalized features*

All the parameters and extensions to our core approach could play the role of features in a machine learning system, which e.g. could learn the weighting of different holing systems or different relations within the same holing system, the pattern type and so on. In this way, the lexical expansions can be tuned towards benefiting a given task at hand. The advantage of using these non-lexicalized features is that a single model can be learned for all expansions, as opposed to one model per language element type (i.e. one classifier per word). Features from the first-order and the second-order representation of our distributional thesaurus have been employed for state-of-the-art lexical substitution in Szarvas *et al.* (2013).

5 CONCLUSION

In this article, we have introduced the new metaphor of two-dimensional text. This metaphor is rooted in structural linguistics, and expands the one-dimensional linear sequence of language elements in a second dimension of associative relations, especially with semantically similar language elements. We have provided a way of operationalizing semantic similarity by splitting syntagmatic observations into terms and context features, and representing them independent of the kind of syntagmatic observation. A scalable, parallelizable implementation of the computation of a distributional thesaurus was laid out in detail. Further, we provide a conceptually simple and efficient method to perform a contextualization of semantic similarity. Overall, our approach constitutes an unsupervised generative model for lexical expansion in context that implements the metaphor of two-dimensional text. In our experiments regarding the quality of distributional similarity, we demonstrated that our pruning method for DT computation is effective: using only the most n significant features per term greatly reduces processing time, and even improves the results. Further, we show that larger corpora lead to higher-quality distributional thesauri, and that we can effectively compute them without relying on lossy compression techniques. Our measure excels over two competitive measures in the literature on very large collections. We have presented a generic method of contextualizing distributional information, which selects entries from the DT entry of the expansion target, and ranks them with respect to their context compatibility. Evaluating our method on the lexical substitution task (McCarthy and Navigli, 2009), we were able to show consistent improvements across all parts of speech, especially for expansion targets with many informing contextual elements. Further, we demonstrated how the two-dimensional expansion can improve part-of-speech tagging without the need to re-train or otherwise alter the tagger. Finally, we laid out a plethora of possible extensions for improving our implementation of the two-dimensional text metaphor. This work is merely a first step towards creating a new, entirely data-driven *model* for computational semantics, as opposed to mere feature-based machine learning or knowledge-intensive approaches.

ACKNOWLEDGEMENTS

This work has been funded by the Hessian research excellence program *Landes-Offensive zur Entwicklung Wissenschaftlich-Ökonomischer Exzellenz (LOEWE)* as part of the *Digital Humanities* research center. We also thank our partners Alfio Gliozzo, Michael Glass and Bonaventura Coppola at IBM Research for contributing to the open source implementation, and for discussions.

REFERENCES

Michele BANKO and Eric BRILL (2001), Scaling to very very large corpora for natural language disambiguation, in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pp. 26–33, Association for Computational Linguistics, Stroudsburg, PA, USA, `http://dx.doi.org/10.3115/1073012.1073017`.

Marco BARONI and Alessandro LENCI (2010), Distributional memory: A general framework for corpus-based semantics, *Computational Linguistics*, 36(4):673–721, ISSN 0891-2017, `http://dx.doi.org/10.1162/coli_a_00016`.

Marco BARONI and Roberto ZAMPARELLI (2010), Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space, in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pp. 1183–1193, Cambridge, Massachusetts, `http://dl.acm.org/citation.cfm?id=1870658.1870773`.

Chris BIEMANN (2009), Unsupervised Part-of-Speech Tagging in the Large, *Research on Language and Computation*, 7(2–4):101–135, ISSN 1570-7075, `http://dx.doi.org/10.1007/s11168-010-9067-9`.

Chris BIEMANN (2010), Co-occurrence cluster features for lexical substitutions in context, in *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-5, pp. 55–59, ISBN 978-1-932432-77-0, `http://dl.acm.org/citation.cfm?id=1870490.1870499`.

Chris BIEMANN and Eugenie GIESBRECHT (2011), Distributional Semantics and Compositionality 2011: Shared Task Description and Results, in *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pp. 21–28, Association for Computational Linguistics, Portland, Oregon, USA, `http://www.aclweb.org/anthology/W11-1304`.

Chris BIEMANN, Uwe QUASTHOFF, Gerhard HEYER, and Florian HOLZ (2008), ASV Toolbox: a Modular Collection of Language Exploration Tools, in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008*, Marrakech, Morocco, `http://www.lrec-conf.org/proceedings/lrec2008/summaries/447.html`.

Chris BIEMANN, Stefanie ROOS, and Karsten WEIHE (2012), Quantifying Semantics Using Complex Network Analysis, in *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, Mumbai, India, `http://aclweb.org/anthology/C/C12/C12-1017.pdf`.

David M. BLEI, Andrew Y. NG, and Michael I. JORDAN (2003), Latent Dirichlet allocation, *Journal of Machine Learning Research*, 3:993–1022, ISSN 1532-4435, `http://dl.acm.org/citation.cfm?id=944919.944937`.

Stefan BORDAG (2008), A comparison of co-occurrence and similarity measures as simulations of context, in *CICLing'08 Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 52–63, Haifa, Israel, `http://dl.acm.org/citation.cfm?id=1787578.1787584`.

Jordan BOYD-GRABER and David M. BLEI (2008), Syntactic Topic Models, in *Neural Information Processing Systems*, Vancouver, British Columbia, `http://www.cs.princeton.edu/~blei/papers/Boyd-GraberBlei2009.pdf`.

Kenneth Ward CHURCH and Patrick HANKS (1990), Word association norms, mutual information, and lexicography, *Computational Linguistics*, 16(1):22–29, ISSN 0891-2017, `http://dl.acm.org/citation.cfm?id=89086.89095`.

Michael COLLINS (2002), Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms, in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing – Volume 10*, EMNLP '02, pp. 1–8, Association for Computational Linguistics, Stroudsburg, PA, USA, `http://dx.doi.org/10.3115/1118693.1118694`.

James R. CURRAN (2002), Ensemble methods for automatic thesaurus extraction, in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing – Volume 10*, EMNLP '02, pp. 222–229, `http://dx.doi.org/10.3115/1118693.1118722`.

James R. CURRAN (2004), *From Distributional to Semantic Similarity*, University of Edinburgh, `http://books.google.de/books?id=2iDbSAAACAAJ`.

Ferdinand DE SAUSSURE (1916), *Cours de linguistique générale*, Payot, Paris, `http://www.bibsonomy.org/bibtex/2e68b895a274b9569189c5ae98db84603/jntr`.

Ferdinand DE SAUSSURE (1959), *Course in general linguistics*, Language (Philosophical Library), Philosophical Library, `http://books.google.de/books?id=FSpZAAAAMAAJ`.

Jeffrey Dean and Sanjay Ghemawat (2004), MapReduce: Simplified Data Processing on Large Clusters, in *Proceedings of Operating Systems, Desing & Implementation (OSDI) '04*, pp. 137–150, San Francisco, CA, USA, `http://doi.acm.org/10.1145/1327452.1327492`.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman (1990), Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, 41(6):391–407, `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.108.8490`.

Inderjit S. Dhillon (2001), Co-clustering documents and words using bipartite spectral graph partitioning, in *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pp. 269–274, ACM, New York, NY, USA, ISBN 1-58113-391-X, `http://doi.acm.org/10.1145/502512.502550`.

Ted Dunning (1993), Accurate methods for the statistics of surprise and coincidence, *Computational Linguistics*, 19(1):61–74, ISSN 0891-2017, `http://dl.acm.org/citation.cfm?id=972450.972454`.

Katrin Erk and Sebastian Padó (2008), A structured vector space model for word meaning in context, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pp. 897–906, Honolulu, Hawaii, `http://dl.acm.org/citation.cfm?id=1613715.1613831`.

Stefan Evert (2005), *The Statistics of Word Cooccurrences: Word Pairs and Collocations.*, Ph.D. thesis, Institut für Maschinelle Sprachverarbeitung, University of Stuttgart, `http://elib.uni-stuttgart.de/opus/volltexte/2005/2371/`.

Eugenie Giesbrecht (2009), In Search of Semantic Compositionality in Vector Spaces, in *Proceedings of the 17th International Conference on Conceptual Structures: Conceptual Structures: Leveraging Semantic Technologies*, ICCS '09, pp. 173–184, Springer-Verlag, Berlin, Heidelberg, ISBN 978-3-642-03078-9, `http://dx.doi.org/10.1007/978-3-642-03079-6_14`.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith (2011), Part-of-speech tagging for Twitter: annotation, features, and experiments, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers – Volume 2*, HLT '11, pp. 42–47, Association for Computational Linguistics, Stroudsburg, PA, USA, ISBN 978-1-932432-88-6, `http://dl.acm.org/citation.cfm?id=2002736.2002747`.

Gene H. Golub and William M. Kahan (1965), Calculating the singular values and pseudo-inverse of a matrix, *Journal of the Society for Industrial and Applied Mathematics: Series B: Numerical Analysis*, 2:205–224, `http://www.citeulike.org/user/rabio/article/2342309`.

James GORMAN and James R. CURRAN (2006), Scaling Distributional Similarity to Large Corpora, in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 361–368, Association for Computational Linguistics, Sydney, Australia, `http://www.aclweb.org/anthology/P06-1046`.

Amit GOYAL, Hal DAUMÉ III, and Graham CORMODE (2012), Sketch Algorithms for Estimating Point Queries in NLP, in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1093–1103, Association for Computational Linguistics, `http://www.aclweb.org/anthology/D12-1100`.

Emiliano GUEVARA (2011), Computing semantic compositionality in distributional semantics, in *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pp. 135–144, Association for Computational Linguistics, Stroudsburg, PA, USA, `http://dl.acm.org/citation.cfm?id=2002669.2002684`.

Zellig S. HARRIS (1951), *Methods in Structural Linguistics*, University of Chicago Press, Chicago, `http://archive.org/details/structurallingui00harr`.

W. Keith HASTINGS (1970), Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, 57(1):97–109, ISSN 1464-3510, doi:10.1093/biomet/57.1.97, `http://dx.doi.org/10.1093/biomet/57.1.97`.

Marti A. HEARST (1992), Automatic acquisition of hyponyms from large text corpora, in *Proceedings of the 14th conference on Computational linguistics – Volume 2*, COLING '92, pp. 539–545, `http://dx.doi.org/10.3115/992133.992154`.

Enrique HENESTROZA ANGUIANO and Pascal DENIS (2011), FreDist: Automatic construction of distributional thesauri for French, in *TALN – 18ème conférence sur le traitement automatique des langues naturelles*, pp. 119–124, Montpellier, France, France, `http://hal.inria.fr/hal-00602004`.

Thomas HOFMANN (1999), Probabilistic latent semantic indexing, in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pp. 50–57, ACM, New York, NY, USA, ISBN 1-58113-096-1, `http://doi.acm.org/10.1145/312624.312649`.

Adam KILGARRIFF, Pavel RYCHLY, Pavel SMRZ, and David TUGWELL (2004), The Sketch Engine, in *Proceedings of EURALEX*, `http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.180.7984`.

Walter KINTSCH (2001), Predication, *Cognitive Science*, 25(2):173–202, ISSN 1551-6709, `http://dx.doi.org/10.1207/s15516709cog2502_1`.

John D. LAFFERTY, Andrew MCCALLUM, and Fernando C. N. PEREIRA (2001), Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, in *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 282–289, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ISBN 1-55860-778-1, `http://dl.acm.org/citation.cfm?id=645530.655813`.

Lillian LEE (1999), Measures of distributional similarity, in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pp. 25–32, College Park, Maryland, ISBN 1-55860-609-3, `http://dx.doi.org/10.3115/1034678.1034693`.

Michael LESK (1986), Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, in *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pp. 24–26, ACM, New York, NY, USA, ISBN 0-89791-224-1, `http://doi.acm.org/10.1145/318723.318728`.

Dekang LIN (1998), Automatic retrieval and clustering of similar words, in *Proceedings of the 17th International Conference on Computational Linguistics – Volume 2*, COLING '98, pp. 768–774, `http://dx.doi.org/10.3115/980432.980696`.

Dekang LIN, Shaojun ZHAO, Lijuan QIN, and Ming ZHOU (2003), Identifying synonyms among distributionally similar words, in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pp. 1492–1493, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, `http://dl.acm.org/citation.cfm?id=1630659.1630908`.

Jimmy LIN and Chris DYER (2010), *Data-Intensive Text Processing with MapReduce*, Morgan & Claypool Publishers, San Rafael, CA, `http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.169.6896`.

Mitchell P. MARCUS, Mary Ann MARCINKIEWICZ, and Beatrice SANTORINI (1993), Building a large annotated corpus of English: the Penn Treebank, *Computational Linguistics*, 19(2):313–330, ISSN 0891-2017, `http://dl.acm.org/citation.cfm?id=972470.972475`.

Marie-Catherine De MARNEFFE, Bill MACCARTNEY, and Christopher D. MANNING (2006), Generating typed dependency parses from phrase structure parses, in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2006*, Genova, Italy, `http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.229.775`.

Diana MCCARTHY and Roberto NAVIGLI (2009), The English lexical substitution task., *Language Resources and Evaluation*, 43(2):139–159, `http://dblp.uni-trier.de/db/journals/lre/lre43.html#McCarthyN09`.

George A. MILLER and Walter G. CHARLES (1991), Contextual correlates of semantic similarity, *Language and Cognitive Processes*, 6(1):1–28, `http://dx.doi.org/10.1080/01690969108406936`.

Tristan MILLER, Chris BIEMANN, Torsten ZESCH, and Iryna GUREVYCH (2012), Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation, in *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pp. 1781–1796, Mumbai, India, `http://aclweb.org/anthology/C/C12/C12-1109.pdf`.

Jeff MITCHELL and Mirella LAPATA (2008), Vector-based Models of Semantic Composition, in *Proceedings of ACL-08: HLT*, pp. 236–244, Columbus, Ohio, `www.aclweb.org/anthology/P08-1028.pdf`.

Sebastian PADÓ and Mirella LAPATA (2007), Dependency-based construction of semantic space models, *Computational Linguistics*, 33(2):161–199, `http://citeseer.uark.edu:8080/citeseerx/viewdoc/summary?doi=10.1.1.86.2026`.

Sebastian PADÓ and Yves PEIRSMAN, editors (2011), *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, Association for Computational Linguistics, Edinburgh, UK, `http://www.aclweb.org/anthology/W11-25`.

Robert PARKER, David GRAFF, Junbo KONG, Ke CHEN, and Kazuaki MAEDA (2011), *English Gigaword Fifth Edition*, Linguistic Data Consortium, Philadelphia, `http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2011T07`.

Ted PEDERSEN, Siddharth PATWARDHAN, and Jason MICHELIZZI (2004), WordNet::Similarity: measuring the relatedness of concepts, in *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL – Demonstrations '04, pp. 38–41, `http://dl.acm.org/citation.cfm?id=1614025.1614037`.

Fernando PEREIRA, Naftali TISHBY, and Lillian LEE (1993), Distributional clustering of English words, in *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, ACL '93, pp. 183–190, Association for Computational Linguistics, Stroudsburg, PA, USA, `http://dx.doi.org/10.3115/981574.981598`.

Reinhard RAPP (2003), Word sense discovery based on sense descriptor dissimilarity, in *Proceedings of the Ninth Machine Translation Summit*, pp. 315–322, `http://www.citeulike.org/user/briordan/article/2911465`.

Matthias RICHTER, Uwe QUASTHOFF, Erla HALLSTEINSDÓTTIR, and Chris BIEMANN (2006), Exploiting the Leipzig Corpora Collection, in *Proceesings of the IS-LTC 2006*, Ljubljana, Slovenia, `http://nl.ijs.si/is-ltc06/proc/13_Richter.pdf`.

Herbert RUBENSTEIN and John B. GOODENOUGH (1965), Contextual correlates of synonymy, *Communications of the ACM*, 8(10):627–633, ISSN 0001-0782, `http://doi.acm.org/10.1145/365628.365657`.

Gerda RUGE (1992), Experiments on linguistically-based term associations, *Information Processing & Management*, 28(3):317 – 332, ISSN 0306-4573, http://www.sciencedirect.com/science/article/pii/030645739290078E.

Magnus SAHLGREN (2006), *The Word-Space Model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces.*, Ph.D. thesis, Stockholm University, http://soda.swedish-ict.se/437/.

Helmut SCHMID (1995), Improvements in Part-of-Speech Tagging with an Application to German, in *Proceedings of the ACL SIGDAT-Workshop*, pp. 47–50, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.52.2255.

Hinrich SCHÜTZE (1993), Word Space, in *Advances in Neural Information Processing Systems 5*, pp. 895–902, Morgan Kaufmann, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.8856.

Hinrich SCHÜTZE (1998), Automatic word sense discrimination, *Computational Linguistics*, 24(1):97–123, ISSN 0891-2017, http://dl.acm.org/citation.cfm?id=972719.972724.

Anders SØGAARD (2011), Semisupervised condensed nearest neighbor for part-of-speech tagging, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers – Volume 2*, HLT '11, pp. 48–52, Portland, Oregon, ISBN 978-1-932432-88-6, http://dl.acm.org/citation.cfm?id=2002736.2002748.

György SZARVAS, Chris BIEMANN, and Iryna GUREVYCH (2013), Supervised All-Words Lexical Substitution using Delexicalized Features, in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-2013)*, Atlanta, GA, USA, http://aclweb.org/anthology/N/N13/N13-1133.pdf.

Ming TAN, Wenli ZHOU, Lei ZHENG, and Shaojun WANG (2012), A scalable distributed syntactic, semantic, and lexical language model, *Computational Linguistics*, 38(3):631–671, ISSN 0891-2017, http://dx.doi.org/10.1162/COLI_a_00107.

Stefan THATER, Hagen FÜRSTENAU, and Manfred PINKAL (2011), Word Meaning in Context: A Simple and Effective Vector Model, in *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 1134–1143, Asian Federation of Natural Language Processing, Chiang Mai, Thailand, http://www.aclweb.org/anthology/I11-1127.

Peter D. TURNEY and Michael L. LITTMAN (2005), Corpus-based Learning of Analogies and Semantic Relations, *Machine Learning*, 60(1-3):251–278, ISSN 0885-6125, http://dx.doi.org/10.1007/s10994-005-0913-1.

Peter D. TURNEY and Patrick PANTEL (2010), From frequency to meaning: vector space models of semantics, *Journal of Artificial Intelligence Research*,

37(1):141–188, ISSN 1076-9757,
`http://dl.acm.org/citation.cfm?id=1861751.1861756`.

Andrew J. VITERBI (1967), Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *IEEE Transactions on Information Theory*, 13(2):260–269, ISSN 0018-9448, doi:10.1109/TIT.1967.1054010,
`http://dx.doi.org/10.1109/TIT.1967.1054010`.

Julie WEEDS (2003), *Measures and Applications of Lexical Distributional Similarity*, Ph.D. thesis, Department of Informatics, University of Sussex,
`http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.538`.

Dominic WIDDOWS and Beate DOROW (2002), A graph model for unsupervised lexical acquisition, in *Proceedings of the 19th International Conference on Computational Linguistics – Volume 1*, COLING '02, pp. 1–7,
`http://dx.doi.org/10.3115/1072228.1072342`.

This work is licensed under the Creative Commons Attribution 3.0 Unported License.
`http://creativecommons.org/licenses/by/3.0/`

Full-fledged temporal processing: bridging the gap between deep linguistic processing and temporal extraction

Francisco Costa and António Branco
University of Lisbon

ABSTRACT

The full-fledged processing of temporal information presents specific challenges. These difficulties largely stem from the fact that the temporal meaning conveyed by grammatical means interacts with many extra-linguistic factors (world knowledge, causality, calendar systems, reasoning). This article proposes a novel approach to this problem, based on a hybrid strategy that explores the complementarity of the symbolic and probabilistic methods. A specialized temporal extraction system is combined with a deep linguistic processing grammar. The temporal extraction system extracts eventualities, times and dates mentioned in text, and also temporal relations between them, in line with the tasks of the recent TempEval challenges; and uses machine learning techniques to draw from different sources of information (grammatical and extra-grammatical) even if it is not explicitly known how these combine to produce the final temporal meaning being expressed. In turn, the deep computational grammar delivers richer truth-conditional meaning representations of input sentences, which include a principled representation of temporal information, on which higher level tasks, including reasoning, can be based. These deep semantic representations are extended and improved according to the output of the aforementioned temporal extraction module. The prototype implemented shows performance results that increase the quality of the temporal meaning representations and are better than the performance of each of the two components in isolation.

Keywords: *temporal processing, temporal extraction, tense, aspect, hybrid approaches, deep linguistic processing, shallow linguistic processing*

Journal of Language Modelling Vol 1, No 1 (2013), pp. 97–154

1 INTRODUCTION

Deep linguistic processing aims at providing grammatical representations of sentences, including their full-fledged semantic representations. This is undertaken by computational grammars whose handcrafted rules encode the regularities uncovered by theoretical linguistics. Deep natural language processing systems have been successfully employed in many applications, like machine translation (Müller and Kasper, 2000; Bond *et al.*, 2005), grammar checking (Bender *et al.*, 2004) and ontology acquisition (Nichols *et al.*, 2006), among others.

While these grammars typically deliver precise linguistic analyses and fine-grained semantic representations of given sentences, they perform less well when it comes to resolving ambiguity and getting at the appropriate representation of a sentence given its context of occurrence. The inverse tension is observed in shallow processing systems. Often resorting to statistical methods, these systems are very helpful at resolving ambiguity, but they perform much worse when it comes to getting at the sophistication of deep semantic representations.

The linguistic expression of time forms a highly intricate semantic subsystem that offers a particularly good illustration of the complementarity between the two approaches and the gap to bridge. Like in any other grammatical dimension, here too ambiguity is pervasive, and each sentence in isolation may bear different temporal readings.

Deep grammars typically handle such proliferation of readings by resorting to some underspecification formalism that allows for its packing. Although this makes it possible to address the efficiency problems associated with this ambiguity, rule-based grammars offer limited means to resolve this ambiguity and to support real-world applications that need to rely on the actual temporal information conveyed by sentences in their contexts.

The area of temporal information extraction, greatly fostered by the TempEval challenges (Verhagen *et al.*, 2007, 2010), has encouraged the development of systems able to extract from texts important pieces of information concerning time. But there is so far little or no exploration of how to combine them with the deep principled semantic representations of the sentences, so that they can help support higher-level temporal processing and reasoning systems. In the opposite direction, much of the sophisticated linguistic information that may be important to improve the accuracy of temporal information extraction is also waiting to be explored.

This paper explores the complementarity of the two approaches, drawing inspiration from other efforts of hybrid natural language processing, such as Crysmann *et al.* (2002) and Frank *et al.* (2003), among others. Our exercise is circumscribed here to the processing of temporal information. A proposal is presented that contributes to an enhanced processing of time by bridging the gap between temporal information extraction and deep linguistic processing.

More specifically, in this paper, we seek to incorporate the temporal information extracted by a system specialized in the processing of temporal information (and developed with machine learning methods) in the meaning representations produced by a deep processing grammar, resulting in semantic representations enriched with more information about time. Our motivation is partly due to the fact that the processing of the expression of time in natural language interacts with a number of extra-linguistic systems (such as calendar systems, or knowledge about the world) that are best handled outside a computational grammar. One example is the processing of temporal expressions, such as *today*, *the fifth of May*, or *two days later*. It may be preferable to compute the exact date that these expressions refer to outside the grammar. Their processing requires access to arithmetic operations (e.g. once the anchor date for the last of these example expressions is determined, it is necessary to add two days to it) and to a calendar system (e.g. so that we know that subtracting two days from March 1, 2012 gets us to February 28, 2012), which grammar formalisms are typically not designed to support.

Many different kinds of information are needed to accurately determine temporal relations conveyed in natural language text. Linguistic knowledge is obviously important, at various levels (lexical, morphological, syntactic, semantic). For instance, aspectual type, which distinguishes various types of eventuality descriptions, such as states, activities, accomplishments and achievements (Vendler, 1967; Dowty, 1979), is partly lexical but also interacts with syntax and semantics, and it features prominently in the semantics literature on the expression of time in natural language. But extra-linguistic knowledge of different sorts also comes into play, such as:

- Pragmatics and knowledge about the world. Relations like causation can override default constraints on interpretation. Like the examples of Lascarides and Asher (1993) show, the chronological order of events can be reflected in the order in which they are presented in text, (1a), but causality relations between the mentioned events can override that preference, (1b):

  (1) a. Max stood up. John greeted him.
      b. Max fell. John pushed him.

- Calendar systems. Time expressions like *next Monday* or *two months earlier* must be interpreted relative to a calendar system, and furthermore there are implicit temporal relations between the referred dates and times that can be be made explicit and explored.
- Logical inference. For instance, temporal precedence is transitive, so sometimes the possible temporal relations conveyed in a piece of text are restricted by what has occurred before (e.g. if event A precedes B, a new event C that precedes A must also precede B).

All these factors are important and should be explored to leverage a temporal extraction system. However, they are difficult to handle in grammar formalisms and grammar development environments.

In this paper, a computational grammar that delivers detailed representations of the meaning of input sentences is extended with a representation of time and aspect, in order to enrich these meaning representations. This extension is based on the linguistic literature on tense and aspect, and it was also developed in such a way that the resulting meaning representations are straightforward to combine with the output of the temporal extractor.

Subsequently, the deep grammar and the temporal extractor are combined, with the purpose of extending and correcting the semantic representations delivered by the grammar as far as temporal meaning is concerned. This combination of the computational grammar with the dedicated temporal extraction system allows the meaning representations produced by the grammar to be improved in the following ways:

- Extending the representations
  It is possible to add further temporal information (that the gram-

mar does not have access to) to the meaning representations output by the grammar. One example is the normalization of temporal expressions (determining the exact date or time that they refer to), which in deep natural language processing systems are often processed separately, for instance by a pre-processing module. In our case, we use the temporal extractor, as it already deals with these expressions, thus avoiding the replication of this functionality in a pre-processing component.

- Specifying the representations
  The meaning representations are in many cases underspecified. When the temporal extractor produces more specific output, the grammar representations can also be made more specific, in accordance with this output.
- Correcting the representations
  Since the grammar only looks at grammatical information, while the temporal extraction system is sensitive to other kinds of information (as hinted at above), it is often more accurate than the grammar in resolving time-related ambiguity. Its output can thus be used to correct meaning representations.

As discussed in this paper, one obtains better and more detailed meaning representations with this combination.

The deep grammar we use is LXGram, presented in Section 2.1. The temporal extractor is LX-TimeAnalyzer, described in Section 2.2.3. The data we use to train LX-TimeAnalyzer, as well as for the evaluation reported here, is TimeBankPT (it is divided into a training set and a test set), which we introduce in Section 2.2.2. The particular systems and data that we used represent the Portuguese language, but the key issues stand for other languages as well: the expression of time and its interface with all these extra-linguistic kinds of knowledge and the meaning representations of time.

We evaluate the performance of this extractor system, the computational grammar, and a combined system that incorporates the two on a common data set. The combined system allows for full-fledged temporal processing and outperforms both the temporal extractor and the deep grammar.

This paper is structured as follows. The next section introduces the key topics that will be dealt with in the remainder of the paper:

deep linguistic processing, hybrid natural language processing, temporal information processing, and the semantics of tense and aspect. The particular systems that we used are also presented. The following sections explain how these elements will work together. Section 3 describes how a deep grammar can be extended to include information about time in the meaning representations that it produces. Section 4 describes and evaluates an approach to integrate the deep processing grammar and the temporal information extractor, and combine their contributions for the processing of the linguistic expression of time. Finally, in Section 5 this article closes with final remarks.

2 BACKGROUND

This section introduces the key elements that will be integrated, with the purpose of combining temporal information extraction and deep semantic representations: a deep grammar that produces such representations (Section 2.1), and temporal information extraction technology, which identifies and normalizes events, dates and times mentioned in a text and classifies temporal relations holding between these entities (Section 2.2). Additionally, we present previous work in the area of hybrid processing (Section 2.3) as well as in the area of the semantics of tense and aspect (Section 2.4).

2.1 *Deep linguistic processing*

Deep linguistic processing grammars associate each input sentence with its grammatical representation, including a representation of its meaning. For the sake of the research exercise reported in this article, LXGram was chosen as the working grammar. LXGram is a deep grammar for Portuguese (Costa and Branco, 2010a).

This grammar is based on the Head-Driven Phrase Structure Grammar (HPSG) grammatical framework (Pollard and Sag, 1994; Sag *et al.*, 2003). HPSG resorts to a unification-based grammatical representation formalism with a type system featuring multiple inheritance and recursive data structures called typed feature structures.

LXGram is implemented in the LKB (Copestake, 2002), an integrated development environment for typed feature structure grammars in general, popular within the HPSG community. The grammar runs on PET (Callmeier, 2000), an efficient parser for HPSG grammars, that allows several input methods, including interfacing with external morphological analyzers, which we make use of. These systems also allow the training and use of a statistical model to discriminate between competing analyses for each sentence (Oepen *et al.*, 2002; Toutanova *et al.*, 2005; Velldal, 2007). This facility is also used with LXGram to rank the parses produced for a given sentence. The grammar outputs all possible parses for a given input sentence, and this model selects the most probable one. The model is trained on CINTIL TreeBank, a treebank obtained by manually selecting the best parse from those produced by the grammar (Silva *et al.*, 2012). Around 2,000 sentences of newspaper text from this treebank were used to train the model.

When run over unrestricted newspaper text, LXGram produces a parse for about 30% of the input sentences, and the disambiguation model correctly identifies the preferred analysis for around 40% of these parsed sentences (Costa and Branco, 2010a). Despite this coverage, we will see below that it already produces very competitive results with temporal processing and results above the state of the art when combined with the shallow temporal extractor.

LXGram explores the core Grammar Matrix system (Bender *et al.*, 2002), which contains a set of implemented grammatical constraints relevant to many languages, following the HPSG framework. It employs Minimal Recursion Semantics (MRS; Copestake *et al.* 2005) as the formalism for the semantic representations it produces.

An important feature of MRS is that it supports underspecified semantic representation. An MRS representation is a tuple containing a global top, a bag of relations labeled with handles and a bag of constraints on handles. Relations labeled with handles are called *elementary predications*, but we will also refer to them as relations in this article. Conjunction is represented by shared labels. Handles can also appear as arguments of these relations, and they are used to represent scope. The main kind of constraint on handles is equality modulo quantifiers ($=_q$), which means that either the two handles are the same handle or one or more quantifier relations (but no relation of a different kind) intervene between the two. They enable the underspecification of the scope between the various relations. An example

MRS representation for the sentence *A black cat can fly* is:[1]

$$
\begin{array}{l}
<h1, \\
\quad \{h2 : \_a\_q(x3, h4, h5),\ h6 : \_black\_a(x3), \\
\quad\ h6 : \_cat\_n(x3),\ h7 : \_can\_v(h8), \\
\quad\ h9 : \_fly\_v(x3)\}, \\
\quad \{h1 =_q h7, h4 =_q h6,\ h8 =_q h9\} >
\end{array}
$$

This representation corresponds to the two scoped formulas that can be obtained from it by scope resolution:

- $\_a\_q(x3,\ \_black\_n(x3) \wedge \_cat\_n(x3),\ \_can\_v(\_fly\_v(x3)))$
  (There is a black cat that possibly flies.)
- $\_can\_v(\_a\_q(x3,\ \_black\_n(x3) \wedge \_cat\_n(x3),\ \_fly\_v(x3)))$
  (It is possible that there is a black cat that flies.)

This is how the scope ambiguity between the existential quantifier and the modal operator is captured. The first reading is obtained when the constraints on the handles are resolved this way: $h1 = h2,\ h4 = h6,\ h5 = h7,\ h8 = h9$. The second one is when $h1 = h7,\ h8 = h2,\ h4 = h6,\ h5 = h9$.

MRS representations are straightforwardly encoded in the typed feature structures manipulated by HPSG grammars. For the sake of readability of this text, we abstain from presenting them in that format.

For the purpose of experimentation, a concrete grammar has to be used. As will be apparent, the solutions put forth are tested with this working grammar but their principles can be easily adapted or transferred to other deep computational grammars delivering an underspecified semantic representation, developed under other grammatical frameworks or for other languages.

Existing computational HPSG grammars typically do not include the meaning representation of tense and aspect in the semantic repre-

[1] We follow the convention of including part-of-speech-inspired labels in the names of the relations in an MRS representation: *n* for relations denoted by nouns, *a* for those related to adjectives and adverbs, *q* in quantifier relations, *v* in verbal relations, etc.

sentations they produce. But because MRSs are used by applications[2] and this sort of information is important even if provided in a very approximate way, a common approach is to enrich the output MRSs with information about grammatical tense and aspect. For instance, the MRS representation for our working sentence *A black cat can fly* often looks like:

$$
\begin{array}{l}
<h1,\\
\quad \{h2 : \_a\_q(x3, h4, h5),\ h6 : \_black\_a(x3),\\
\quad\ h6 : \_cat\_n(x3),\ h7 : \_can\_v(e10\{tense\ present\}, h8),\\
\quad\ h9 : \_fly\_v(e11, x3)\},\\
\quad \{h1 =_q h7, h4 =_q h6,\ h8 =_q h9\} >
\end{array}
$$

Here, two event variables have been added to the relations for *can* and *fly*, an approach similar to that of Davidson (1967). These event variables can have features of their own. The one for *can* has a *tense* feature with the value *present*. This is an indication of the verb tense used in the verb form corresponding to this relation.

This approach, which is common to several existing computational HPSG grammars, has the disadvantage of mixing semantic information with grammatical information. This mixing is undesirable, because semantic representations are supposed to explicitly describe truth conditions, which grammatical categories fail to do. The motivation for our work is also to eliminate grammatical information from semantic representations, as far as tense and aspect are concerned.

2.2 *Temporal information extraction*

There is a long research tradition on extracting the information about time that is conveyed in natural language text. Some recent evaluation campaigns have given it more attention, as they focused precisely on this task. They include TempEval (Verhagen *et al.*, 2007), TempEval-2 (Verhagen *et al.*, 2010), and TempEval-3 (UzZaman *et al.*, 2013). Besides encouraging work on the topic, the TempEval campaigns have provided data that can be and has been explored to develop and evaluate systems that automatically annotate natural language text with the temporal information they convey.

[2] Machine translation is one application where MRS representations have been extensively used, in this case as the level to which transfer rules apply (Flickinger *et al.*, 2005; Nygaard *et al.*, 2006; Nichols *et al.*, 2007).

<s>*In Washington* <TIMEX3 tid="t53" type="DATE" value="1998-01-14">*today*</TIMEX3>*, the Federal Aviation Administration* <EVENT eid="e1" class="OCCURRENCE" stem="release" aspect="NONE" tense="PAST" polarity="POS" pos="VERB">*released*</EVENT> *air traffic control tapes from* <TIMEX3 tid="t54" type="TIME" value="1998-XX-XXTNI">*the night*</TIMEX3> *the TWA Flight eight hundred* <EVENT eid="e2" class="OCCURRENCE" stem="go" aspect="NONE" tense="PAST" polarity="POS" pos="VERB">*went*</EVENT> *down.*</s>
<TLINK lid="l1" relType="BEFORE" eventID="e2" relatedToTime="t53"/>
<TLINK lid="l2" relType="OVERLAP" eventID="e2" relatedToTime="t54"/>

Figure 1: Simplified sample of the annotations in TempEval for the fragment: *In Washington today, the Federal Aviation Administration released air traffic control tapes from the night the TWA Flight eight hundred went down*

These data are annotated with an annotation scheme similar to TimeML (Pustejovsky *et al.*, 2003a). Figure 1 shows a small, simplified extract of the data from the first TempEval challenge, with TimeML-style annotations.

The words that denote events are annotated using `EVENT` tags. An example is the word referring to the event of the FAA's releasing of the tapes. `EVENT` tags are also employed to annotate words denoting states (such as the situations denoted by verbs like *love* or *want*). For this reason, in this context the terms *event*, *situation*, and *eventuality* are employed interchangeably in this paper, to refer to states and events. This use of the term *event* is common in the literature on temporal extraction.

The `TIMEX3` tags surround temporal expressions, such as *today*. In this working example, the temporal expression *today* denotes the date normalized as `1998-01-14`. The attribute `value` of `TIMEX3` elements holds this normalized representation.

The `TLINK` elements at the end describe temporal relations between events and dates, times or other events. For instance, the event of the plane going down is annotated as temporally preceding the date denoted by the temporal expression *today*.

The first two TempEval challenges had as their main tasks the automatic identification of the temporal relations. That is, the value of the `relType` attribute of the `TLINK` elements (such as the ones in Figure 1) had to be determined, and all other annotations were given. Temporal relation classification is also the most interesting problem in temporal information extraction. The other tasks that are necessary to automatically annotate text with TimeML (identifying and normalizing temporal expressions and events) show better evaluation results, and they also have a longer research history.

TempEval featured three tasks: A, B and C.[3] Task A was about classifying the temporal relation that holds between an event and a time mentioned in the same sentence (they could however be far apart in the sentence, as the temporal relation represented by the `TLINK` with the `lid` with the value `l1` in Figure 1). Task B focused on the temporal relation between events and the document's creation time, which is also annotated in TimeML (not shown in that figure). Task C was about classifying the temporal relation between the main events of two consecutive sentences. The goal of all these tasks was to determine the type of a given temporal relation. The possible values for the type of relations are `BEFORE`, `AFTER` and `OVERLAP`, as well as `BEFORE-OR-OVERLAP`, `OVERLAP-OR-AFTER` and `VAGUE`, but the last three values occur very infrequently in the annotated data that were made available for TempEval.

2.2.1 State of the art in temporal information extraction

Table 1 shows a synopsis of the results of the first two TempEval competitions, taken from Verhagen *et al.* (2009, 2010), for the main tasks of classifying temporal relations. The data used in these two competitions are similar but not identical, hence the different baselines.

This table does not show the results of TempEval-3, because they are so difficult to compare to previous work: (i) the training data set used is substantially larger (twice the size), (ii) the evaluation setup is different (in TempEval-3, the temporal relation classification tasks are performed from raw text; in the first two TempEval competitions, the remaining gold annotations were given to participants), (iii) the inventory of relation types is different, (iv) and the evaluation measure is also different – the temporal awareness score of UzZaman and Allen (2011) is used instead of classification accuracy.

[3] TempEval-2 had additional tasks, about identifying and normalizing events and temporal expressions. It also had an additional temporal relation classification task, about pairs of events mentioned in the same sentence. Furthermore, the names of the tasks in TempEval-2 are different. We use the names employed in TempEval.

Table 1:
Results for English
in TempEval

		Task A	Task B	Task C
TempEval	Best system	**0.62**	**0.80**	**0.55**
	Avg. of all participants	0.56	0.74	0.51
	Majority class baseline	0.57	0.56	0.47
TempEval-2	Best system	**0.65**	**0.82**	**0.58**
	Avg. of all participants	0.61	0.78	0.53
	Majority class baseline	0.55	0.59	0.49

Table 2:
Results for English in TempEval-2: temporal expressions and events (F-measure for extent recognition and accuracy for the attributes)

Temporal expressions					
	Extents	`type`	`value`		
Best system	**0.86**	**0.98**	**0.85**		
Avg. of all participants	0.78	0.86	0.57		
Median	0.82	0.91	0.59		
Events					
	Extents	`class`	`tense`	`aspect`	`polarity`
Best system	**0.83**	**0.79**	**0.92**	**0.98**	**0.99**
Avg. of all participants	0.74	0.72	0.75	0.97	0.99
Median	0.79	0.77	0.86	0.97	0.99

TempEval-2 also evaluated the recognition of temporal expressions and events (i.e. identifying their extents in text) and their normalization (filling in the various attributes of the `EVENT` and `TIMEX3` elements visible in Figure 1). A synopsis of the results is in Table 2. The averages of all participants reported in this table are affected by a few extremely low scores; therefore we also show the median values.

The several systems that participated in the first two TempEval challenges resorted to different methods. There were symbolic solutions as well as machine learning approaches. Different levels of linguistic analysis, ranging from shallow processing, such as POS-tagging, to full syntactic parsing, were explored as a means to provide information used in rules or as classifier features. This variety of approaches can also be seen amongst the best systems of TempEval-2. The TRIPS and TRIOS systems (UzZaman and Allen, 2010) used a combination of parsing and machine learning methods such as conditional random fields (Lafferty *et al.*, 2001) and Markov logic networks

(Richardson and Domingos, 2006). TIPSem (Llorens *et al.*, 2010a) also used conditional random fields trained using several kinds of features, including features extracted from the output of a syntactic parser, namely that of Charniak and Johnson (2005) for English. Like UzZaman and Allen (2010), the NCSU systems (Ha *et al.*, 2010) employed Markov Logic using features taken from different natural language processing tools. Ha *et al.* (2010) gave a bigger emphasis to features that capture lexical relations between the event terms involved (such as similarity relations between *producing* and *creating* events, antonymy relations between the terms *open* and *close*, etc.).

TimeML, the TimeBank (Pustejovsky *et al.*, 2003b) – a TimeML annotated corpus which served as the basis for the data used in TempEval – and the TempEval challenges have been very influential in the area of temporal information extraction. The work of Denis and Muller (2010) offers a comparison of the set of temporal relations considered in TimeML and other temporal algebras developed earlier, namely those of Allen (1983, 1984) and Bruce (1972).

Also, a lot of recent work has used the TimeBank and the data sets made available in the two TempEval challenges. Verhagen and Pustejovsky (2008) present a system that automatically annotates raw text with TimeML, including annotations for events, time expressions and temporal relations. Chambers *et al.* (2007) trained machine learning classifiers on the TimeBank, namely Naïve Bayes (John and Langley, 1995) classifiers. They were concerned with temporal relations between pairs of events, which could be in the same sentence or not. Their system's goal intersects Task C of the first TempEval challenge (relations between events in different sentences). Their algorithm operates on two stages. In the first stage, they try to learn some properties of the events in the temporal relation, such as tense, grammatical aspect and aspectual class. Here they use some morpho-syntactic features as well as features based on information provided by WordNet (Fellbaum, 1998). In the second stage, they classify the temporal relation between those events. They use as classifier features the information obtained in the first stage, as well as other kinds of features based on the syntactic structure of the sentences where the events are mentioned. Llorens *et al.* (2010b), similarly to Llorens *et al.* (2010a), explore the contribution of semantic role labeling to temporal information processing.

Machine learning methods have become dominant in addressing the problem of extracting the temporal ordering of what is described in a text. One major limitation of machine learning methods is that they are typically used to classify temporal relations in isolation, and therefore it is not guaranteed that the resulting temporal ordering is globally consistent. Yoshikawa *et al.* (2009) and Ling and Weld (2010) seek to overcome this limitation using Markov logic networks, which learn probabilities attached to first-order formulas. Some of the participants of the second TempEval used a similar approach (Ha *et al.*, 2010; UzZaman and Allen, 2010). Denis and Muller (2011) cast the problem of learning temporal orderings from texts as a constraint optimization problem. They search for a solution using Integer Linear Programming (ILP), similarly to Bramsen *et al.* (2006), and Chambers and Jurafsky (2008). Because ILP is costly (it is NP-hard), the latter two only consider *before* and *after* relations. Rather than classifying a temporal relation between two time intervals, Denis and Muller (2011) and Lee (2010) classify four relations between four instants (the endpoints of the two original time intervals). Symbolic or hybrid approaches have also been used. This was the case of the WVALI system (Puşcaşu, 2007), one of the participants of the first TempEval competition and the one with the best results for some of the tasks.

The logical properties of temporal relations make temporal information processing stand out from many of the other natural language processing tasks. UzZaman and Allen (2011) propose a new way to evaluate temporal information processing systems. Instead of the usual precision and recall metrics used in the first two TempEval competitions, they argue that it is better to compute the temporal closure of the reference annotations and confront the result with a system's output. This is because a system may identify temporal relations that are not part of the reference annotations but nevertheless are logical consequences of the ones that are in fact annotated.

Despite the prominence of the TimeML annotated data sets mentioned earlier (the TimeBank and the data sets of the TempEval challenges) and the plethora of work using them, there are further resources with temporal annotations. One is the WikiWars corpus (Mazur and Dale, 2010). Its scope is more limited than that of the TimeBank and the data used in the TempEval challenges, because it is annotated only for temporal expressions, leaving out events and temporal relations. The kind of task it supports is thus similar to the early efforts of the Temporal Expression Recognition and Normalization evaluation (Ferro *et al.*, 2004) and the previous Message Understanding Conferences (MUC-6, 1995; MUC-7, 1998), where a concerted effort for the annotation of time expressions first took place. Yet another corpus of English featuring temporal annotations (Bethard *et al.*, 2007) contains annotated temporal relations between events denoted by words in a specific syntactic relation (one heads the clause that is the complement of the other one).

Work on the topics of temporal expression recognition (identifying the boundaries of temporal expressions in text) and normalization (assigning each of them a normalized representation of the time or date that they refer to) has produced quite good results for some time now (Negri and Marseglia, 2004; Strötgen and Gertz, 2013; Angeli *et al.*, 2012; Llorens *et al.*, 2012). Still, in recent years, the topics of temporal expression recognition and normalization have not been abandoned. WikiWars, just mentioned, is a recent corpus where time expressions are annotated. Other recent work on this topic includes that of Zhao *et al.* (2010). Additionally, there has been interest in new problems related to temporal expressions. Kolomiyets *et al.* (2011) investigate the portability of time expression recognition to non-newswire domains, since most of the annotated data consist of news articles (the exception being WikiWars). Their idea is to generate additional training examples by replacing temporal expression words with potential synonyms, taken from WordNet and other similar resources. This technique potentially increases the number of word types seen in training as part of a time expression.

2.2.2 Data with annotations about time

The data released in the first TempEval challenge were for English only. The second TempEval challenge released data for Chinese, English, French, Italian, Korean and Spanish (although only English and Spanish attracted participants to the competition). Since then, efforts to manually annotate temporal phenomena have continued for several languages (Pustejovsky and Stubbs, 2011; Xue and Zhou, 2010; Zhou and Xue, 2011), and a number of corpora featuring similar temporal annotations have been developed for several languages: Chinese

Table 3:
Size of the data set

	Train	Test
Sentences	2,281	351
Word tokens	60,782	8,920
Annotated events	6,790	1,097
Annotated temporal expressions	1,244	165
Annotated temporal relations		
Task A	1,490	169
Task B	2,556	331
Task C	1,735	258
Total	5,781	758

(Cheng *et al.*, 2008), French (Bittar *et al.*, 2011), Italian (Caselli *et al.*, 2011), Korean (Im *et al.*, 2009), Romanian (Forăscu and Tufiş, 2012).

For Portuguese, there is the TimeBankPT corpus (Costa and Branco, 2010b, 2012d). This corpus is an adaptation of the original TempEval data to Portuguese, obtained by translating it and then adapting the annotations. The two corpora – TimeBankPT and the original English data set used in the first TempEval challenge – are quite similar (Costa and Branco, 2012d), but the languages are of course different.

TimeBankPT is used here to train and evaluate the temporal information extraction component. Just like the original English corpus for TempEval, it is divided into a training part and a testing part. The original English corpus is composed of news documents. Many of these documents are taken from the Wall Street Journal, and they belong to the domain of economics. TimeBankPT is thus also composed of documents of this genre and domain. Some figures pertaining to the size of this data set are presented in Table 3.

2.2.3 LX-TimeAnalyzer

For the experiments reported in the present paper, an independent temporal extractor is used. It is called LX-TimeAnalyzer (Costa and Branco, 2012b,c) and annotates raw text with temporal annotations. These annotations are similar to the ones used in the first two TempEval challenges, based on TimeML (Pustejovsky *et al.*, 2003a), and illustrated in Figure 1 above. LX-TimeAnalyzer annotates raw text with events, temporal expressions, and temporal relations. This system runs on Portuguese input text, and it was trained with the data just presented above in Section 2.2.2.

In order to produce these annotations, several tasks are performed: (i) identifying temporal expressions and events mentioned in the text; (ii) normalizing these time expressions (annotating the `value` attribute of `TIMEX3` elements, where the date or time referred to by the temporal expression is recorded in a standardized format); (iii) filling in the values of the remaining attributes of the `EVENT` and `TIMEX3` elements that were recognized; (iv) identifying temporal relations, i.e. which pairs of entities (events and times) should be linked with temporal relations; and (v) classifying these temporal relations (overlap, precedence, etc.).

Most of these tasks are performed with machine learning classifiers trained on the training data of TimeBankPT. The tasks of normalizing temporal expressions and identifying temporal relations are performed by handcrafted rules, and most of the annotated attributes of `EVENT` elements are directly based on the output of other natural language processing tools, namely a part-of-speech tagger and morphological analyzer. The classifiers used to identify event terms and temporal expressions use features based on information that also comes from these tools (part-of-speech, lemma, inflectional features) and a context window of two words on each side of the target word. There is a dedicated machine learning classifier for the attribute `class` of `EVENT` terms.

The normalization of temporal expressions makes use of Joda-Time 2.0,[4] which implements calendar systems as well as many operations between dates (e.g. it can calculate that two days after February 28, 2013 is March 2, 2013).

The models that classify temporal relations are produced with machine learning classifiers that use several features that capture many types of information. These features are numerous, and for this reason it is not possible to provide a full account of them, which is presented in (Costa, 2013). Briefly, there are:

- Superficial features based on information from a part-of-speech tagger (e.g. the conjunction nearest the event that enters the temporal relation under classification);

[4] `http://joda-time.sourceforge.net`

- Features that encode information about logical inferences. For instance, we solve task B before the other two tasks, and sometimes information about task B temporal relations as well as the implicit temporal relations between the times and dates mentioned in the text can constrain the temporal relations in the other subsequent tasks;
- Fine-grained information about aspectual type. TimeML makes a distinction between states and non-states in the attribute `class` of `EVENT` elements. We explore a more fine-grained distinction, as we make use of four aspectual types, following the work of Vendler (1967) and Dowty (1979) (as well as the large body of literature that follows them) more closely;
- Information about the world (e.g. a verb like *predict* typically precedes in time what is predicted, but a verb like *report* typically follows in time what is reported).

Crucially, LX-TimeAnalyzer makes use of several pieces of extra-linguistic information, such as the logical constraints between temporal relations (when classifying temporal relations) or calendar systems (when normalizing temporal expressions), that are typically not available to a deep natural language processing system. Depending on the formalism employed in the implementation of a deep grammar, it may not even be feasible or practical to implement this kind of knowledge in such a system. It certainly is not possible in the LKB, where LXGram is implemented, but even if it were, there is still the question of whether it would be appropriate to encode extra-linguistic information in a deep grammar.

Evaluation results show that LX-TimeAnalyzer performance is at the level of the state-of-the-art for English (Costa and Branco, 2012b,c), except for the task of event detection (determining whether a given word token denotes an event). This problem is somewhat hard for nouns. The best system to identify events in the second TempEval resorted to, among other things, WordNet (Llorens *et al.*, 2010a), an approach that is not available for Portuguese currently, as there is no WordNet for this language with the breath and maturity of the English WordNet. This makes event identification harder for Portuguese (Costa and Branco, 2012c). Table 4 presents the evaluation results for LX-TimeAnalyzer, using the test data of TimeBankPT. The evaluation

Temporal expressions	Score	Events	Score	Temporal relations	Score
Extents	0.85	Extents	0.72	Task A	0.67
`type`	0.91	`class`	0.74	Task B	0.80
`value`	0.81	`tense`	0.95	Task C	0.55
		`aspect`	0.96		
		`polarity`	0.99		

Table 4: Performance of LX-TimeAnalyzer on the test data of TimeBankPT

measures reported in that table are the F-measure for the problems of identifying the extents of event terms and temporal expressions and accuracy for the remaining tasks. These results are very similar to the state of the art for English (cf. Table 1 and Table 2).

2.3 *Hybrid natural language processing*

The present paper follows a hybrid approach to natural language processing.

Within the HPSG community, we find, among others, the work of Adolphs *et al.* (2008), which allows the grammars developed in the LKB (presented above in Section 2.1) to see the output of shallow tools as AVMs (Attribute-Value Matrices, the data structures that HPSG grammars manipulate). This work builds on previous efforts to combine shallow and deep processing with HPSG, like the work of Crysmann *et al.* (2002) and Frank *et al.* (2003). Frank *et al.* (2003) combines a deep grammar with a shallower parser, resulting in efficiency gains of a factor of 2.25. Crysmann *et al.* (2002) additionally use shallow morphological analysis, part-of-speech tagging and named entity recognition to guess information about unknown words (words not in the lexicon of the deep grammar). This results in an increase in grammar coverage from 12.5% to 22.1%, on a corpus of 20,000 newspaper sentences.

Similar work is that of Schäfer (2006), who develops a software architecture designed to combine shallow and deep systems, with the purpose of making the deep systems more robust. The author shows that this approach increases the efficiency and the coverage of the deep system by a factor of more than two. Since then, hybrid techniques such as these have become popular within deep processing. LXGram uses a similar approach, where morphological information output by shallow tools is used to enable the grammar to process unknown words (though we do not use a shallow parser to improve efficiency).

Grover and Lascarides (2001) is an earlier work that also uses the morphological information coming from shallow tools to increase the robustness of a computational grammar, namely when it comes to dealing with out-of-vocabulary words.

In the Verbmobil project (Wahlster, 2000) on speech-to-speech translation, multiple parsers are used to aid machine translation. Several of them are run in parallel (a symbolic HPSG grammar, a statistical parser, and a chunker). They produce meaning representations in a common format. When the parsers fail to provide analyses that fully span an utterance, the fragments that they produce are combined, resulting in an analysis for the entire utterance (Rupp *et al.*, 2000).

Also in the context of the Verbmobil project, particularly relevant to our work is that reported in Alexandersson *et al.* (2000) and Stede *et al.* (1998). They extract mentions of times and dates from the semantic representations produced by the parsers and employ a specialized module to map these semantic representations to a canonical representation of these dates and times. Their work shows that recognizing temporal expressions can be done with a parser. However, like us, they consider that other problems, like this problem of temporal expression normalization, are best handled with external technology. In our work, where an existing and stand-alone temporal extraction system is available, it is not necessary to have the grammar recognize temporal expressions, since the extraction system (which must be used to normalize them anyway) already performs this task.

Within the Lexical-Functional Grammar (LFG) framework (Kaplan and Bresnan, 1982), Brun (1998) describes a pre-processing step where nominal multiword expressions as well as time expressions are recognized in the input that is to be subsequently parsed by a grammar. Named entity recognition has also been integrated in this pre-processing stage in several computational LFG grammars (Kaplan *et al.*, 2004; Butt *et al.*, 1999).

The approach we present in this paper is also a hybrid approach, where a deep grammar is combined with shallower tools. But in our case we combine information of a different kind. We are interested in putting together different methods to extract temporal relations from text: with the deep processing grammar, which looks exclusively at grammatical information, and with a dedicated temporal extraction system, which has access to extra-linguistic knowledge. Instead of using external tools to pre-process the grammar's input, we use the output of a tool specialized in temporal extraction to refine the grammar's output in a post-processing step. In our scenario, post-processing is preferred to pre-processing because: (i) the additional information that is being brought to the grammar is directly about meaning (i.e. it is about temporal relations and representations of the times denoted by time expressions); and (ii) the time expressions recognized and annotated by the temporal extraction system do not necessarily correspond to syntactic constituents.[5]

2.4 *The semantics of tense and aspect*

There is a vast body of linguistic literature on the semantics of tense and aspect. Our implementation of tense and aspect in the deep grammar, described below in Section 3, is inspired by previous work that we briefly describe in this section.

Davidson (1967) is the first author to reify events. In HPSG, this approach has been popularized in a number of analyses, including Sag *et al.* (2003), as well as in several HPSG implementations, like the English Resource Grammar (Flickinger, 2000) and the Grammar Matrix (Bender *et al.*, 2002). A survey of the advantages over the alternatives can be found in Kamp and Reyle (1993, pp. 504–10).

Reichenbach (1947) described tenses as temporal relations between several pairs of times, not just an event time and an utterance time (or speech time). In particular, he introduced the concept of a reference time that mediates the relation between those two times. This idea has been maintained in subsequent work by other authors.

[5] The system recognizes time expressions according to the TIMEX3 specification (Saurí *et al.*, 2006). Many TIMEX3 elements are syntactic constituents (for instance, many are noun phrases), but some elements of noun phrases (such as relative clauses) are left out of the annotated extents of these elements annotated with TIMEX3 tags, as the inclusion of such elements would make a parser necessary to determine these extents. If the annotated time expressions always corresponded to syntactic constituents, this information could be exploited in order to contrain the parser's search space. As they do not, there is no benefit in detecting them in a pre-processing step.

Some influential ideas originating in Discourse Representation Theory (DRT), of Kamp and Reyle (1993), have also crept into many analyses of tense. This is the case in the observation that past tense denotes overlap of the event time with a past time in the case of stative situations but inclusion in the case of non-stative situations.

Intricately related to tense is aspect. A large body of literature exists on this topic, with the work of Vendler (1967) and Dowty (1979) being seminal.

Pustejovsky (1991) posits a separate level of representation for the event structure associated with predicates and their arguments and advocates the decomposition of events into sub-events. For instance, a sentence like *the door closed* is analyzed as a process (*the door closing*) followed by a state (*the door is closed*). This is similar in spirit to the work of Moens and Steedman (1988).

In the framework of HPSG, Van Eynde (2000) develops an analysis for the Dutch tenses and temporal auxiliaries inspired by DRT in its semantic aspects. The work of Yoshimoto and Mori (2002) combines HPSG with a DRT analysis of tense. Bonami (2002) is an HPSG analysis of aspect shift inspired by the work of de Swart (1998, 2000). This phenomenon is treated by positing implicit aspectual operators, which we also resort to. Flouraki (2006) focuses on aspectual constraints on the various tenses of Modern Greek, modeling them with HPSG. Relevant to our work is also that of Goss-Grubbs (2005), which develops an analysis of tense and aspect for English using MRS. This work encodes aspectual type by typing event variables, and it also resorts to positing explicit aspectual operators in the semantic representations. It does not make use of explicit temporal relations or the various Reichenbachian times (reference time, speech time, etc.); instead it encodes tense as a feature of time variables.

Bobrow *et al.* (2007) is also similar work, inasmuch as it is about a computational system that produces meaning representations of its input which contain non-trivial information about time. In its representations, the system includes explicit temporal relations between events and the speech time. It does not, however, include information about aspect or make use of reference times.

3 DEEP PROCESSING OF TENSE AND ASPECT

A semantic representation for tense and aspect was implemented in the grammar that was presented above in Section 2.1, taking into account the possibility of it being extended with additional information relevant to time coming from temporal information extraction systems.

The grammar was extended with an implementation of tense and aspect inspired by much of the literature just referred to above. The following running example illustrates the various aspects of the implementation:

(2) *A atriz mudou-se de França para os Estados Unidos em*
the actress moved from France to the United States in
fevereiro de 1947.
February of 1947
The actress moved from France to the United States in February 1947.

The MRS representation for this sentence, as produced by the grammar, is shown in Figure 2. Temporal information can be seen in the *is-before* and *at* relations, that relate the event time $t9$ with the utterance time $t10$, and aspectual information can be seen in the *aspectual-operator* relations as well as the feature *culmination*, which indicates that the associated eventuality (the *moving* event) contains a culmination as one of its sub-events (i.e. it is a culmination or a culminated process).

The remainder of this section provides more details on the implementation of tense and aspect in the working grammar, and how they are reflected in the meaning representations such as the one in Figure 2.

3.1 *Tense*

It is important to distinguish between grammatical tense and semantic tense: we will use the first expression to refer to inflectional morphology alone, and the second one to refer to the temporal and aspectual meaning they convey.

Each predicate denoted by a verb, adjective, preposition or adverb receives a Davidsonian semantic representation, with an event

$$
\begin{array}{l}
<h1, \\
\quad \{h3 : \_o\_q(x4, h5, h6), \\
\quad\; h7 : \_atriz\_n(x4), \\
\quad\; h8 : at(e2\ \{culmination\ +\}, t9), \\
\quad\; h8 : is\text{-}before(t9, t10\ \{t\text{-}value\ utterance\text{-}time\}), \\
\quad\; h8 : aspectual\text{-}operator(e2, e12, h11), \\
\quad\; h11 : \_mudar\_v(e12, x4), \\
\quad\; h11 : \_de\_p(e14, e12, x13), \\
\quad\; h15 : proper\_q(x13, h16, h17), \\
\quad\; h18 : named(x13,\ \text{“França”}), \\
\quad\; h11 : \_para\_p(e20, e12, x19), \\
\quad\; h21 : \_o\_q(x19, h23, h22), \\
\quad\; h24 : named(x19,\ \text{“Estados Unidos”}), \\
\quad\; h11 : \_em\_p(e26, e12, x25), \\
\quad\; h27 : udef\_q(x25, h28, h29), \\
\quad\; h30 : \_fevereiro\_n(x25), \\
\quad\; h30 : \_de\_p(e31, x25, x32), \\
\quad\; h33 : proper\_q(x32, h34, h35), \\
\quad\; h36 : named(x32,\ \text{“1947”})\}, \\
\quad \{h1 =_q h8,\ h5 =_q h7,\ h16 =_q h18,\ h23 =_q h24,\ h28 =_q h30, \\
\quad\; h34 =_q h36\} >
\end{array}
$$

Figure 2: MRS for *A atriz mudou-se de França para os Estados Unidos em fevereiro de 1947* “The actress moved from France to the United States in February 1947”

variable as its first argument. This variable is not explicitly quantified, but assumed to be bound by an existential quantifier. This is in line with a substantial amount of the HPSG literature, including computational implementations such as the English Resource Grammar (Flickinger, 2000) and the Grammar Matrix (Bender *et al.*, 2002). An example is the predicate *_mudar_v* (for the verb form corresponding to English “move”) in Figure 2: its first argument ($e12$) is an event variable.

Additionally, an *at* relation pairs this event variable with a temporal index: in Figure 2 this relation is labeled with $h8$ and relates the event variable $e2$ with the temporal index $t9$. This temporal index represents the event time. In the existing literature on tense, some authors use quantified time variables, while other authors use free time variables. Partee (1973) presents arguments for a free variable approach. Our temporal indices are compatible with this approach. Temporal indices have their own type in the grammar, and a feature T-VALUE is appropriate for this type. This feature locates the index in the time line.

Depending on the grammatical tense, there are then temporal relations between temporal indices, in the spirit of Reichenbach, who also describes tense as temporal relations between various times.

In our example, the Portuguese verb is in the *pretérito perfeito* tense. The semantics of this tense is ambiguous between a simple perfective past (i.e. the situation occurred in the past and is culminated) and a present perfect (the situation has a resulting state that holds and is relevant at the present). The event time is before the utterance time and, accordingly, there is a temporal relation *is-before* with the event time as its first argument.

This particular example is an adaptation of a Reichenbachian representation, where one would expect two time relations (the event time is simultaneous with a reference time and this reference time precedes the utterance time). Our option to diverge in this particular case is motivated by the ambiguity of grammatical tenses like the *pretérito perfeito*. This grammatical tense is ambiguous with respect to semantic tense, viz. the simple past (which has the Reichenbachian analysis just mentioned) and the present perfect (where the event time precedes the reference time, and the reference time is simultaneous with the utterance time). Since it is not possible to underspecify this distinction in the semantic representations, there are two options: duplicate the number of analyses provided by the grammar for each verb with this tense in the input (this is the approach of Van Eynde 2000 for Dutch, but it is computationally costly and does not seem justifiable as both representations essentially describe a past event); or use a simplified representation that covers both interpretations. We chose the second route, arriving at what has just been described.

With other tenses, the grammar delivers representations resorting to reference times.

The second argument of the temporal relation *is-before* is another temporal index, $t10$, with a T-VALUE specified to have the value *utterance-time*. This is how the speech time is represented. According to what has been presented so far, the relevant representation fragment is thus:

$at(e2,\ t9) \wedge$ *is-before*$(t9,\ t10\ \{$*t-value utterance-time*$\}) \wedge$
_mudar_v$(e2,\ x4)$

That is, the event described by the form of the verb *mudar* “move” occurred in a time that precedes the utterance time.

It is thus worth noting that grammatical tense presents two levels of ambiguity that must be resolved:

- The same form can correspond to more than one grammatical tense. An English example is the verb form *put*, which can, for instance, be present tense or past tense. Portuguese also contains similar ambiguities, e.g. forms like *corremos* (“we run” or “we ran”).
- The same grammatical tense can cover more than one meaning when it comes to locating a situation in time. An English sentence like *I leave tomorrow* shows that present tense can refer to the future. Usually this tense locates an event in the present. Portuguese has similar cases.

This two-fold ambiguity is accounted for by a two-layer analysis in the working grammar. The first layer consists of a set of rules that map surface form to grammatical tense. The second layer consists of a set of rules that map grammatical tense to semantic representations of tense. Both are implemented as lexical rules, i.e. unary rules that apply to single lexical items (verb forms in this case).[6]

[6] One example is the following. In Portuguese, present tense can be frequently used with a future meaning, although of course it can also be used to refer to a present situation. This possibility exists in English, too (e.g. *The train leaves tomorrow*). With this organization in two layers, a present tense verb form is analysed in the following fashion. A rule in the first layer is responsible for the morphology: it maps between the lemma of the verb, which is what is encoded in the grammar’s lexicon, and the actual surface form. It also produces a morphological representation in which the grammatical tense of this verb form is encoded, in a dedicated feature. In the second layer, two rules can apply. One of them associates present tense morphology with present semantics. It adds to the meaning representation for the sentence where the verb form occurs that the situation denoted by the verb holds at a time that overlaps the speech time. The second rule

In the case of the rules in the first layer, the orthographic form of their output is different from that of their input (one is the dictionary form of the word, as that is what is listed in the grammar's lexicon, and the other one is an inflected form). The rules in the second layer do not change the spelling of their input. When we combine the grammar with an external morphological analyzer, the second layer of rules is still applied in the grammar, but the application of the rules in the first layer is dependent on the annotations coming from the external analyzer.

3.2 *Aspect*

Aspectual type is accounted for with the help of three Boolean features: *culmination* (positive for culminations and culminated processes), *process* (positive for processes and culminated processes) and *state* (positive for states). This representation is intended to capture the proposal of Moens and Steedman (1988), who decompose a culminated process into a process followed by a culmination. In our representation the two features *process* and *culmination* would be positive, which indicates that this culminated process is composed of two sub-events: a process and a culmination (although the order in which they occur is not made explicit in our representation). These three features are appropriate for event variables.

Even though aspectual type is also a lexical property, it is difficult to annotate it (Pustejovsky *et al.*, 2006). In our implementation, we abstain from recording aspectual type in the lexicon. This would require the annotation of a large part of the existing lexicon, which already contains several thousands of lexical entries. Another difficulty is that aspectual type depends on word sense, which is typically not dealt with by deep grammars, including LXGram.

However, contextual (i.e. syntactic) constraints on aspect are indeed implemented. These are represented by aspectual operators, which are functions from situation descriptions to situation descriptions, and they appear as relations in the MRS representations.

that can possibly apply to a morphological present is one that encodes future semantics, expanding the meaning representation with a temporal precedence relation between the speech time and the time at which the situation denoted by the verb holds.

For instance, we represent a function from state descriptions to culmination descriptions as *aspectual-operator*(e_2 {*culmination* +}, e_1 {*state* +}, X). Here, e_1 is a state, e_2 is a culmination, and X is the MRS representation for the state e_1. The event variable of the resulting situation (e_2 in this example) is included in the representation. We also make use of an extra argument, which is just a pointer for the event variable of the argument (e_1 in this example), as this is useful when post-processing MRS representations.

We follow Bonami (2002) in assuming that all aspectually sensitive relations allow for at most one implicit aspectual operator. These implicit aspectual operators account for aspectual coercion. Therefore every context that allows aspectual coercion must introduce either zero or one aspectual operators in the semantic representation: zero if no aspectual coercion actually occurs, or one otherwise.

Because it is not possible to underspecify the number of relations in an MRS, one *aspectual-operator* is introduced in every aspectually sensitive context, although in general it is not specified which operator it is (in line with Bonami 2002). That is, one underspecified operator is always introduced. We assume that sometimes it stands for a dummy relation (i.e. the identity function), in the cases when no aspectual shift occurs.

Several elements are sensitive to aspectual type. Tense is one of them. Consider the two example sentences below. They correspond to the English sentence *Samuel liked that wine.*

(3) a. O Samuel gostou desse vinho.
b. O Samuel gostava desse vinho.

The difference between the two is grammatical tense, but they also convey different temporal and aspectual meanings. In the first one the verb is in the *pretérito perfeito*, discussed above. In the second one the verb is in the *pretérito imperfeito*. Both are past tenses, but the first is perfective whereas the second one is imperfective.

Perfective aspect constrains the whole event to be telic (a culmination or a culminated process). Imperfective aspect constrains it to be a state in Portuguese. The first sentence means that Samuel liked the wine at some point in the past, but he no longer does. It may suggest a particular wine tasting episode that has ended (i.e. he liked the wine that he drank at some specific time in the past, as in the English sentence *Samuel enjoyed that wine*), or it may mean that for some time Samuel liked that (kind of) wine, but he no longer does. The second one cannot be about a particular episode. It says that Samuel used to like that kind of wine, and he may still like it.

The grammar assigns to the first sentence a semantic representation expressing this:

$$\begin{aligned}&\textit{at}(e\ \{\textit{culmination}\ +\},\ t)\ \wedge\\&\textit{is-before}(t,\ t2\ \{\textit{t-value utterance-time}\})\ \wedge\\&\textit{aspectual-operator}(e,\ e2,\ \textit{gostar}(e2,\ X)),\end{aligned}$$

where X is the representation for the verb's arguments.

This representation is similar to the one presented above in the discussion about tense, but it includes information about aspect as well. In particular, an *aspectual-operator* was added scoping over the relation for the main verb in this sentence. This operator is introduced in the semantics by the lexical rule responsible for semantic tense (together with the temporal relations seen in this MRS fragment), as tenses impose aspectual constraints at the clausal level (Bonami, 2002). The constraint that the event variable e be telic (its feature *culmination* has the value $+$) also comes from the *pretérito perfeito* tense.

By contrast, the second sentence receives a representation like:

$$\begin{aligned}&\textit{at}(e\ \{\textit{state}\ +\}, t)\ \wedge\ \textit{overlaps}(t,\ t2)\ \wedge\\&\textit{is-before}(t2,\ t3\ \{\textit{t-value utterance-time}\})\ \wedge\\&\textit{aspectual-operator}(e,\ e2,\ gostar(e2,\ X)),\end{aligned}$$

where X is the representation for the verb's arguments.

The *pretérito imperfeito* conveys a different temporal meaning, and therefore the temporal relations in the semantic representation are different. This tense does not indicate that the associated situation no longer holds at present, and accordingly the associated temporal relations are more vague with respect to the relation between the event time t and the utterance time $t3$. Unlike the *pretérito perfeito* tense, which introduces an aspectual operator that produces telic situations, the *pretérito imperfeito* constrains the whole clause to be a state. In this example, this is encoded in the event variable e, with its feature *state* constrained to have the $+$ value.

The verb *gostar* "like", instantiating the third argument of the *aspectual-operator* relation, is a state. Even though lexical aspect is not encoded in the grammar (and therefore there is no restriction on the aspectual features of $e2$) for the reasons mentioned above, our encoding of aspect at the syntactic level, as was just illustrated, is important because it can capture distinctions such as the one illustrated by this pair of sentences.

Additionally, it can be straightforwardly extended with lexical aspect: if we knew that "like" is lexically a state, then the *aspectual-operator* in the second sentence is a function from states to states (i.e. it is the identity function, and does not change the basic meaning of the verb). The aspectual operator in the first sentence would be a function from states to telic situations. This causes a shift in meaning, as a culmination is added, corresponding to the end of the underlying state. As mentioned above, there can be two results: Samuel's liking of that kind of wine ended in the past, or the situation is associated with a specific episode that similarly ended in the past.

The implementation of aspect in the grammar interacts with many elements that are sensitive to aspect: many verbs, which impose aspectual constraints on their complements (some examples are the progressive auxiliary, which combines with processes, but also verbs like *stop* and *finish*); durational adverbials (*for* adverbials, which combine with processes, and *in* adverbials, which combine with culminated processes, are widely studied with respect to this phenomenon); tenses (as just briefly illustrated); etc.

A full description of the semantics of all tenses implemented in the grammar is outside the scope of this paper and would be tedious, but an example with the present tense can also be presented. A sentence like *O Samuel gosta desse vinho* "Samuel likes that wine" receives an MRS representation along the following lines:

$$\begin{array}{l} at(e\ \{state\ +\}, t)\ \wedge \\ includes(t,\ t2\ \{t\text{-}value\ utterance\text{-}time\})\ \wedge \\ aspectual\text{-}operator(e,\ e2,\ gostar(e2,\ X)), \end{array}$$

where X is the representation for the verb's arguments.

Here t is the event time, and $t2$ is the utterance time. The present tense is assumed to be an imperfective tense, similar to the past imperfective tense mentioned above: it is associated with an overlap relation, and constrains the clause where it occurs to describe a state. We follow DRT in further assuming that semantic present is special in that this overlap relation is more specific than just overlap, and it is an inclusion relation: the event time includes the utterance time. Because the verb *gostar* “like” is a state lexically, this is another example where the aspectual operator involved is the identity function.

3.3 *Backshift*

There is also an implementation of backshift, or sequence of tense, in this grammar. The pairs of English sentences in (4), adapted from Michaelis (2006), illustrate this issue, which is visible in indirect speech. Each sentence in parentheses is the direct speech counterpart of the embedded clause in the same line, and yet they often (but not always) show different tenses. For instance, the example in (4b) shows an embedded past tense that corresponds to a present tense form in the direct speech utterance.

(4) a. Debra said she **likes** wine. (“I like wine”)
  b. Debra said she **liked** wine. (“I like wine”)
  c. Debra said she **brought** the wine. (“I brought the wine”)
  d. Debra said she **had brought** the wine. (“I brought the wine”)
  e. Debra said she **will bring** some wine. (“I will bring some wine”)
  f. Debra said she **would bring** some wine. (“I will bring some wine”)

The example in (5), from Rodríguez (2004), clearly shows that in syntactic contexts such as the one exemplified by these sentences, tense can be interpreted relatively. In (5) the past tense that occurs in the embedded clause (i.e. in *drank*) is associated with a verb that describes a situation that in the most natural reading for this sentence will occur in the future. Here, past tense merely indicates precedence with respect to the situation mentioned in the matrix clause, through the use of a future construction (*will tell*). In other words, this past tense form is interpreted relative to another mentioned event rather than with respect to the speech time.

(5) María will tell us after the party tomorrow that she drank too much.

The data are essentially identical for Portuguese as far as backshift is concerned. Further analysis can be found in Costa and Branco (2012a). The implementation of backshift in the grammar follows the analysis proposed in that paper.

The grammar makes use of the machinery of HPSG (unification, multiple inheritance and recursive data structures called typed feature structures) to implement constraints on the various tenses such that some of them are always interpreted relative to the speech time whereas others can be interpreted relative to the speech time or the event time of a higher verb, depending on the syntactic context where they occur.

The implementation accounts for cases like the examples in (4). An embedded present tense conveys an overlap temporal relation between the time of the eventuality described in the embedded clause and the speech time, as exemplified in (4a). An embedded future is similarly interpreted relative to the speech time, but conveying a precedence relation between the speech time and the time of the embedded eventuality (4e). An embedded past tense can be associated with an overlap relation or with a precedence relation between the time of the eventuality in the embedded clause and the time of the eventuality mentioned in the higher clause, as in (4b) and (4c). Constructions similar to the English past perfect, as in (4d), trigger a temporal precedence relation between the time of the eventuality mentioned in the embedded clause and the time of the eventuality in the main clause. Finally, sentences similar to the one in (4f) are associated with a precedence temporal relation between the time of the main event and the time of the embedded event.

4 FULL-FLEDGED TEMPORAL PROCESSING

This section describes how the information output by a temporal extraction system can be integrated with the deep semantic representations produced by the grammar.

4.1 *Integration of deep processing and temporal extraction*

The temporal extraction system outputs information that can be combined with the semantic representations delivered by the grammar, resulting in semantic representations enriched with more and better information about time. In some cases, it is preferable to compute these pieces of temporal information outside the grammar; in other cases it is not even possible to compute them in the grammar. One such example is the normalization of temporal expressions, which, as explained above in Section 1, requires access to arithmetic operations and to a calendar system. Deep grammars are implemented with specialized description formalisms and, in some cases, in platforms that do not even make arithmetic operations available.[7]

Typically, those specialized grammatical formalisms have a number of characteristics: they are developed exclusively with grammatical modeling in mind and often do not support operations that are not directly needed for this modeling; the formalisms used in handcrafted grammars are typically categorical (they let one say whether a sentence is either grammatical or ungrammatical, not whether it is better or worse than an alternative), thus making it difficult to represent gradient or statistical information; and, since computational efficiency is an important concern for these systems, many are very restrictive.[8] Another characteristic of computational grammars is that their context is limited, as they typically only look at one sentence at a time. Because of this, they do not have access to information present in other parts of the document, which temporal extraction systems can take advantage of.

The expression of time in natural language and its meaning representation make particularly strong cases where these limitations can be felt. These tasks deal with a number of aspects that require extralinguistic knowledge and as such are difficult or even impossible to implement in their full breadth in these specialized formalisms. Among

[7] This is the case of LXGram and all grammars implemented in the LKB. The LKB accepts a language called TDL – Type Description Language (Krieger and Schäfer, 1994) – which has no support for arithmetic. By contrast, modern programming languages make arithmetic operations available, and it is possible to find for them good implementations of calendar systems. For the implementation of the temporal extractor described above in Section 2.2.3, Joda-Time 2.0 (`http://joda-time.sourceforge.net`) was used, which provides many calendar operations as well as many operations on time intervals.

[8] For instance, the LKB, where LXGram is developed, is very fast, but, for efficiency reasons, does not allow the direct encoding of many kinds of constraints that are standard in the HPSG literature (Melnik, 2005).

```
<TIMEX3 tid="t0" functionInDocument="CREATION_TIME" value="2012-01-10T15:00:00"/>
<s>A atriz <EVENT eid="e5">mudou</EVENT>-se da França para os Estados Unidos em
<TIMEX3 value="1947-02" tid="t15">fevereiro de 1947</TIMEX3>.</s>
<TLINK lid="l2" eventID="e5" relType="BEFORE" relatedToTime="t0"/>
<TLINK lid="l3" eventID="e5" relType="OVERLAP" relatedToTime="t15"/>
```

Figure 3: Example text with (simplified) temporal annotations. The English translation is *The actress moved from France to the United States in February 1947.*

these aspects we find: (i) arithmetic and calendar systems (for the normalization of temporal expressions, as just mentioned); (ii) reasoning (temporal relations have several logical properties that can be exploited, such as the transitivity of temporal precedence); (iii) the modeling of world knowledge and pragmatics (where statistical information about what is usual or expected may constitute important heuristics to determining the chronological order of the described situations); etc.

In particular, it is possible to augment these semantic representations output by the grammar in the following ways:

- Extending the representations
  It is possible to add to the MRS representations output by the grammar further temporal information that the grammar does not have access to.
- Specifying the representations
  The MRS representations are in many cases underspecified, and in some such cases they can be made more specific.
- Correcting the specifications
  The temporal extraction system is sensitive to both grammatical and extra-grammatical information. It is often more accurate in resolving time-related ambiguity than the grammar, which considers grammatical features only. As such, the extractor's output can be used to correct the MRS representations produced by the grammar.

The following paragraphs provide details on how these three aspects are handled by our system that combines the deep grammar and the temporal extractor. To that end we return to our running example, presented above in (2) and repeated below for convenience:

(2) *A atriz mudou-se de França para os Estados Unidos em*
the actress moved from France to the United States in
fevereiro de 1947.
February of 1947

The actress moved from France to the United States in February 1947.

The temporal annotation obtained by the temporal extraction system for this running example is displayed in Figure 3. That example shows two annotated temporal relations, namely an overlap relation between the moving event and the month of February 1947, and a temporal precedence relation between this event and the document creation time.

The semantic representation obtained by the grammar for this example is shown in Figure 2 on page 120. The objective is thus to enrich the grammar-derived representation by exploring the temporal annotations shown in Figure 3.

4.1.1 Extending the MRS representations

The outcome of this combination is presented in Figure 4. As can be seen by comparing Figures 2, 3 and 4, there are several pieces of information that are incorporated into the resulting MRS representation. These additions are highlighted in bold in Figure 4.

The first one is the information about the document's creation time (the `TIMEX3` element in Figure 3). Temporal extraction systems register when a document was created (in our example this is `"2012-01-10T15:00:00"`), which can be determined from meta-data or with heuristics. This information can be incorporated in the MRS representations, specifying the utterance time. The normalized value for the document's creation time is used to fill in the `T-VALUE` of the temporal index for the utterance time. In Figure 4, this is the temporal index $t10$.

The second type of information to add is about temporal expressions. An argument is added to the relation for the head word of that expression that was identified as a temporal expression by the extraction system. This argument is instantiated with a temporal index whose *t-value* feature contains the normalized representation of the time expression. In our example, the temporal expression *fevereiro*

$$
\begin{aligned}
&<h1,\\
&\{h3 : \_o\_q(x4, h5, h6),\\
&\quad h7 : \_atriz\_n(x4),\\
&\quad h8 : at(e2\ \{culmination\ +\}, t9),\\
&\quad h8 : is\text{-}before(t9, t10\ \{t\text{-}value\ \text{“}\mathbf{2012\text{-}01\text{-}10T15:00:00}\text{”}\}),\\
&\quad h8 : aspectual\text{-}operator(e2, e12, h11),\\
&\quad h11 : \_mudar\_v(e12, x4),\\
&\quad h11 : \_de\_p(e14, e12, x13),\\
&\quad h15 : proper\_q(x13, h16, h17),\\
&\quad h18 : named(x13,\ \text{“França”}),\\
&\quad h11 : \_para\_p(e20, e12, x19),\\
&\quad h21 : \_o\_q(x19, h23, h22),\\
&\quad h24 : named(x19,\ \text{“Estados Unidos”}),\\
&\quad h11 : \_em\_p(e26, e12, x25),\\
&\quad h27 : udef\_q(x25, h28, h29),\\
&\quad h30 : \_fevereiro\_n(x25, \mathbf{t69}\ \{t\text{-}value\ \text{“}\mathbf{1947\text{-}02}\text{”}\}),\\
&\quad \mathbf{h30} : \mathbf{overlaps}(\mathbf{t9}, \mathbf{t69}),\\
&\quad h30 : \_de\_p(e31, x25, x32),\\
&\quad h33 : proper\_q(x32, h34, h35),\\
&\quad h36 : named(x32,\ \text{“1947”}\},\\
&\{h1 =_q h8,\ h5 =_q h7,\ h16 =_q h18,\ h23 =_q h24,\ h28 =_q h30,\\
&\quad h34 =_q h36\} >
\end{aligned}
$$

Figure 4: Final MRS for *A atriz mudou-se de França para os Estados Unidos em fevereiro de 1947* “The actress moved from France to the United States in February 1947”

de 1947 “February 1947” is originally given the MRS representation:

$$
\begin{aligned}
< h27,\ \{\ & h27 : udef\_q(x25,\ h28,\ h29),\\
& h30 : \_fevereiro\_n(x25),\\
& h30 : \_de\_p(e31, x25, x32),\\
& h33 : proper\_q(x32, h34, h35),\\
& h36 : named(x32,\ \text{“1947”})\ \},\\
\{\ & h28 =_q h30,\ h34 =_q h36\ \} >.
\end{aligned}
$$

An extra argument is added to the *_fevereiro_n* relation (with the label $h30$), filled with a temporal index containing the normalized value for the temporal expression, as shown in Figure 4: $< h30 :$ *_fevereiro_n*($x25$, $t69$ {*t-value* "1947-02"}) >.[9]

Finally, additional temporal relations detected by the temporal extraction system are incorporated in the MRS.

The only temporal relations originally present in the MRS representations are the ones directly related to verb tense, since the grammar only looks at grammatical information. These are always between an event and the utterance time or the event of the higher clause in the case of backshift phenomena (Costa and Branco, 2012a).

But temporal information systems can extract more temporal relations than those. These extra relations can be added to the MRS representations. In our example this is the *overlaps* relation between the event time *t9* of the moving event and the temporal index *t69* for the time conveyed by the temporal expression *fevereiro de 1947* "February 1947" : $< h30 :$ *overlaps*($t9, t69$) >.

[9] The resulting representation is somewhat redundant, and we believe it can be improved. However, this issue is far from trivial, although it may seem so at first. The intuitive alternative would be to replace the entire material in the original MRS for this temporal index. In this example, the five relations (and the two handle constraints) for the expression *fevereiro de 1947* "February 1947" would be completely eliminated from the MRS and replaced by a temporal index. This temporal index would occur as the second argument of the *_em_p* relation, for the preposition corresponding to English *in*: *_em_p*($e26, e12, t69${*t-value* "1947-02"}). This alternative has two problems that must be noted.

The first one is illustrated by a sentence like *2007 saw the birth of the iPhone*. Here, a temporal expression occurs as the subject of a verb. With the intuitive representation, the first argument of the predicate for the verb *to see* would end up being a temporal index. This seems wrong, as the first argument of that predicate would not be of the expected type.

The second problem is related to examples like *that awful year*. This is a time expression that includes material (namely the adjective *awful*) that is not present in the normalized value of the temporal expression (which would just consist of a number representing a calendar year). Replacing the entire MRS representation of this noun phrase for a temporal index would create a representation that does not include all the information present in the analyzed input sentence.

We believe that the problem of adequately modeling the semantic representation of temporal expressions is an interesting question for linguistics to further clarify, for these reasons. As such, an admittedly simplistic solution was chosen in our integrated representation.

To implement the integration of the original MRS produced by the grammar with the information coming from the temporal extraction system, all that is needed is an alignment between the word tokens in the original text and the semantic relations that correspond to those tokens. In our experimental setup, this is achieved quite straightforwardly since the PET parser, the parsing engine used with the grammar, allows the grammar to provide character spans next to each relation in the output MRS representations. These character spans describe the character positions of the linguistic material corresponding to that relation and are used for the alignment and merging of the deep temporal representations with the temporal relations extracted.

4.1.2 Increased semantic specification

The temporal relations identified by the grammar can be made more specific on the basis of the output of the temporal extractor. One example illustrating this is related to the following sentence, taken from the training data of TimeBankPT, with the original English sentence also presented below in italics:

(6) Esperava-se que Bush autorizasse os comandantes navais a usar "a mínima força necessária" para interditar os navios de carga para o Iraque e a partir do Iraque, disse um oficial americano.
Bush was expected to authorize naval commanders to use "the minimum force necessary" to interdict shipments to and from Iraq, a U.S. official said.

TimeBankPT (and the English data set used in TempEval) contains TimeML annotations for this sentence describing temporal relations between the document's creation time and several events, namely those represented by *esperava-se* "it was expected", *usar* "use", and *disse* "said". Similarly, the temporal extractor is capable of identifying these temporal relations.

The temporal semantics implemented in the grammar also encodes several temporal relations between situations described by finite verb forms and the speech time, which is similar to the document's creation time. However, in some cases, these semantic representations are less specific than the TimeML annotations.

A case in point is the imperfective past tense in indirect speech contexts, which is exemplified in this sentence with the verb form *esperava* "was expected". Here the semantics will encode that the event conveyed by the embedded *esperava* overlaps the one conveyed by *disse* "said". This is as expected, because this tense is associated with these kinds of readings in this context.[10] This semantic representation does not say anything about the relation between the embedded situation and the speech time or document's creation time. This is not a shortcoming of the implemented grammar; it is what is justified from the point of view of the linguistic analysis. But this information is readily available in the output of the temporal extractor, and therefore can be incorporated in the final MRS representation.

Another case that is not trivial to treat in the grammar alone is verb forms in the conditional mood. The grammar implementation assigns them a future of past interpretation: the described event occurs at a time that follows another time that precedes the speech time. Therefore, the direct relation between events introduced by verb forms in this tense and the speech time is not available in the MRS representation produced by the grammar, and in fact can be any one.

In the annotated data, however, there can be cases of temporal annotations between events introduced by verbs in the conditional and the document's creation time.

4.1.3 Corrections to the temporal representations

In some cases, the temporal extraction system can be used to correct the MRSs output by the grammar.

In cases of conflict between the initial temporal relations identified by the grammar and the ones given by the temporal extractor, the initial representations produced by the grammar can be corrected if the temporal relations identified by the extractor are considered more reliable than the ones that the grammar produces.

[10] "Past under past" constructions (Comrie, 1986; Declerck, 1990; Hornstein, 1991; Abusch, 1994; Michaelis, 2011) may be ambiguous in English. For example, in *John said he was ill* the two situations described can be simultaneous, but in *John said he fell down* the one described by the embedded verb precedes the one in the matrix clause. In Portuguese, the two interpretations are distinguished by the past tense used: the imperfective past is used in the former case, and the perfective past is used in the latter one (Costa and Branco, 2012a).

This is because the grammar only looks at grammatical tense, whereas the temporal information system takes other features into account, and can identify cases where grammatical tense is insufficient or misleading. An example of this is the case of the historical present, that is, the grammatical present being used to describe a past event, such as in the sentence *In 1939 Germany invades Poland*. This is an important property of our proposal.

Another example where corrections are fruitful is also connected to the use of present tense in Portuguese. English allows this tense to be used to describe future events, as in *The train leaves tomorrow*. In Portuguese this is much more pervasive, and because of that each occurrence of this tense is given this reading, as well as a present reading, by the grammar. The representations for the two different readings (present and future) are not underspecified (because they have different aspectual constraints, i.e. they constrain the three Boolean features that we use to encode aspect, as presented above, differently). Rather, each occurrence of this grammatical tense is ambiguous between present and future, triggering two distinct analyses. As mentioned before, the system uses a statistical model to discriminate between competing analyses for each sentence. By causing the analysis to branch out in these cases, the choice of present vs. future is determined by this parse selection model.

Not surprisingly, as far as this distinction goes, this parse selection model performs quite poorly when compared to a dedicated temporal annotation system, as shown in the next section. That is, there are several cases when the best interpretation given by the grammar erroneously assigns future semantics to present tense verb forms or vice versa. In these cases, the integration component corrects the final MRS representation by changing the temporal relations there so that said representation is in accordance with the output of the temporal extractor.

4.2 *Evaluation*

A test suite of sentences exemplifying the phenomena that the grammar should be able to deal with was created. It contains sentences in the various tenses, sentences with forms of the auxiliary *ter* "have" combining with a past participle, sentences with a progressive construction similar to the English construction composed of *be* and an

-ing form, sentences with forms of *ir* "go" with an infinitive (similar to English "going to" constructions), and sentences featuring adverbs like *hoje* "today", *ontem* "yesterday", and *amanhã* "tomorrow", which feature different combinatorial possibilities with the different tenses. This test suite is used for regression tests during grammar development and contains 38 sentences. The grammar is able to correctly parse all of these sentences and provides correct temporal representations for them.

The test suite is useful to check for bugs in the implementation and ensure that the expected results are seen, but it might not be representative of what is seen in practical scenarios. So an evaluation with unseen data was conducted.

Evaluating this approach presents specific challenges. There is no gold-standard available with MRS annotations that contains temporal information similar to what is presented here. And in fact, it is quite difficult to produce MRS representations manually, as they contain many re-entrancies. For these reasons, we resort to manual evaluation. Since the temporal extractor was developed using the training set of TimeBankPT, the test part of this corpus is unseen and can be used for evaluation of the integrated solution as well.

To this end, the 20 documents comprising the test portion of TimeBankPT were parsed with the grammar. On large corpora of native Portuguese text taken from newspapers and the Wikipedia, the grammar is capable of analyzing around ⅓ of all sentences (Costa and Branco, 2010a), as already mentioned above in Section 2.1. In the present case, 24% of the sentences in the test set of TimeBankPT got a parse.[11] Since the integration of the grammar with the extractor is not meant to increase the coverage of the former, the sentences that receive no parse were left out of this evaluation exercise. There remained 84 sentences in the test set.

This section provides evaluation results for the several tasks directly involved in the integration of the grammar with the temporal extraction system. First, the recognition and normalization of tempo-

[11] We assume that this lower coverage is due to the fact that many of the documents composing this data set are taken from the Wall Street Journal (as TimeBankPT is a translation of the English corpus used in TempEval), and there was no effort to have the grammar deal with text from the financial and economic domains, which contain quite a number of syntactic idiosyncrasies.

ral expressions is discussed. This task is performed by the temporal extractor and then combined with the MRS representations output by the grammar, as discussed above. Here the results for the integrated output are thus the same as those for the temporal extractor.

After that, evaluation results are presented for two problems that are similar to the Tasks A and B of TempEval discussed above. Since the temporal extractor identifies events and temporal expressions and temporal relations between these, and these temporal relations are added to the MRS representations, the performance of the extractor and that of the integrated system are discussed. Finally, evaluation results are provided for the classification of temporal relations between events and the speech time or the document's creation time (i.e. Task B of TempEval). In this respect both the grammar and the temporal extractor are evaluated in isolation, since each can output these temporal relations. The integrated system, which corrects the MRS representations with the information coming from the extractor, is also evaluated.

The Task C of TempEval is not used by our integrated approach. Since Task C relates events mentioned in different sentences, a discourse representation is necessary to combine them in an informed way. This is not something that the typical deep linguistic technology does at the moment.[12]

Table 5 summarizes the results discussed in the rest of this section and obtained on the parsed sentences of the test data of TimeBankPT. In this table, n/a marks results that are not available, as the grammar is not intended to perform the corresponding tasks.

4.2.1 Evaluation of temporal expression recognition and normalization

Since the integrated system enriches the original MRS representations with representations for the temporal expressions that occur in the underlying text, this dimension was evaluated.

As mentioned above, we restricted our attention to the sentences for which there was a parse produced by the grammar. We looked at all temporal expressions that can be found in these sentences. The system was evaluated with respect to two factors. First, we want to know

[12] An exception is Boxer (Curran *et al.*, 2007), which can handle some cross-sentential phenomena, such as pronoun resolution and presupposition.

	Total	Grammar	Extractor	Combined system
Temporal expressions	32			
Recognition		n/a	28/32 (88%)	28/32 (88%)
Normalization		n/a	27/32 (84%)	27/32 (84%)
Event – time pairs	44			
Task A		n/a	25/44 (57%)	25/44 (57%)
Finite verbs	111			
Task B		83/111 (75%)	92/111 (83%)	104/111 (94%)

Table 5: Accuracy of the grammar, the temporal extraction system and the combined system for several tasks (% correct)

how many temporal expressions are recognized correctly. Second, we want to know how they are normalized, since these normalized values appear in the final representations.

Temporal expressions are somewhat infrequent and, in these 84 sentences, only 32 such expressions occur. Of these, 88% are recognized correctly. The remaining ones are either not recognized at all or their boundaries are not identified correctly. 84% are recognized correctly and also normalized correctly (or 96% of the ones that are recognized correctly). From the point of view of normalization, the difficult cases are very vague ones such as *the night*. These cases fail to be normalized and as such are not incorporated in the final MRS representations.

Although some of the temporal expressions occurring in this data set fail to be recognized and incorporated in the final MRS representations, the ones that are indeed inserted there are almost all correctly normalized (96%).

4.2.2 Evaluation of temporal relations between mentioned times and events

As mentioned above, the final MRS representations also include temporal relations between the times and dates and the events mentioned in the input sentences, since these relations are delivered by the temporal extractor (cf. Task A).

These temporal relations occurring in the semantic representations of the parsed sentences were checked for correctness. There are only 44 such relations, because only a few sentences contain multiple temporal expressions and multiple events. 57% of these relations are correctly encoded. A considerable number of the errors occur when the times and events being related are mentioned very far apart in the sentence or the syntactic relationship between the expressions denoting them is not direct. If we restrict our attention to pairs of events and times that are mentioned in the same clause, this score goes up to 68%.

Since the grammar provides us with this information, we are considering only adding these temporal relations to the MRS representations in these cases when the relevant expressions occur in the same clause. So even though temporal information processing technology still has a considerable amount of error, to some extent we can at least increase precision by sacrificing recall in a straightforward way if this is considered preferable.

4.2.3 Evaluation of temporal relations with the speech time

One final aspect to evaluate is how many of the temporal relations between events and the speech time or document's creation time, output by the final integrated temporal processing system, are correct. This is similar to the Task B of TempEval.

The grammar assigns temporal relations to events and states represented by finite forms of verbs only, for the reasons already mentioned. TimeBankPT includes annotations also for events denoted by words of other parts-of-speech, most importantly nouns. Even though the extractor can also identify these, it is not as accurate in doing so, as mentioned above. For this reason, the integrated system does not expand MRS representations with temporal information for events that are not given by verbs, and likewise we also ignore them in this evaluation.

For each sentence, only the preferred parse output by the grammar, as determined by the parse selection model, is considered. The grammar produced a correct output for 75% of all temporal relations between the situations described in these parsed sentences and the document's creation time/speech time.

As mentioned above, one difficulty is assigning the correct meaning to present tense verb forms. As they are ambiguous between future and present semantic values and this distinction is chosen by a general parse selection model, it is rarely the case that it is correctly resolved. The temporal extractor is much better at this particular problem, as it employs several features that are relevant to it. For instance, aspectual type is very relevant; depending on the language, the future interpretation of present tense is much harder or even impossible with stative verbs (Van Eynde, 1998, p. 249). The grammar has no information about lexical aspect, but the extractor has some, in the form of the aspectual indicators as well as the features `class` and even `stem` (since this is a lexical property). This problem accounts for 56% of the errors produced by the grammar for this task. Other errors were less interesting and had a smaller impact overall.

The temporal extractor gets 83% of these temporal relations between finite verb forms and the speech time/document's creation time right, better than the 75% of the grammar. The largest source of error has to do with identifying events: many of the verbs for which the grammar produces temporal relations are not recognized as events by the temporal extractor, and therefore no relation is posited for them. Note that TimeML does not annotate verbs used in generic statements (such as *Lions are mammals*) as events, and furthermore the annotations for event terms that occurred fewer than 20 times in the English data used in TempEval were removed. Therefore the training data of TimeBankPT, which is also used to train the event identification model used in LX-TimeAnalyzer, contains many examples of verbs that are not annotated as being event terms.[13]

The system combining the output of the grammar and that of the temporal extractor delivers temporal relations between finite verbs and the speech time/document's creation time with 94% accuracy. This is a better result than either the grammar (75%) or the temporal extractor (83%) in isolation.

Overall, these results show that integrating a specialized temporal extractor with a deep grammar can be fruitful in practice in increasing the quality of the temporal meaning representations and the accuracy of the resulting system.

[13] As a side note, if one removes these cases and looks only at those that were identified by both the grammar and the temporal extractor, the success rate of the latter in classifying the temporal relation with the document's creation time goes up to 97%. This is substantially better than the results presented above for the task B of TempEval because here we are looking exclusively at events denoted by verbs, which are easier to order with respect to the utterance time than those given by words with a different part-of-speech.

5 CONCLUSIONS

This article presents a novel contribution to the processing of the linguistic expression of time in deep natural language processing systems by combining them with data-driven methods. As interpreting the temporal ordering of the events mentioned in a text is indeed affected by phenomena that are difficult to model in a symbolic system, like knowledge of the world, machine learning methods can capture the contribution of factors whose impact is not well understood. To this end, it was discussed how to combine the outcome of temporal information extraction technology with the semantic representations produced by a deep processing grammar.

This combination helps to resolve the ambiguity preserved in the underspecified semantic representation. One very important point is that it also allows for the representations produced by deep grammars to encode extra-linguistic information – e.g. the normalized representation of the speech time – that is relevant to interpret these representations but hard to obtain with these grammars alone.

Finally, with the present contribution towards full-fledged temporal processing, this paper adds to the overall discussion and quest on how to make progress in natural language processing by means of hybrid systems that combine the complementarity of the symbolic and probabilistic approaches in a way that their strengths can be amplified and their shortcomings mitigated. The resulting system presents better performance than each of the two components in isolation, both quantitatively (as measured in terms of accuracy) and qualitatively (as it outputs truth-conditional representations of the meaning of sentences that includes but is not limited to information about time).

Future work is needed to address temporal relations between events mentioned in different sentences. In this respect, there is some work on the temporal structure of discourse, also using HPSG. One example is the work of Hitzeman *et al.* (1995), although in some cases this specific proposal leaves these temporal relations underspecified. It would be interesting to check how proposals such as this one compare with current temporal relation classification technology for the task C of the first TempEval challenge. Future work can check this by implementing a similar solution with the grammar.

Future work can address other ways to combine the two subsystems (the extractor and the grammar). The integration can also work in the direction opposite of the one explored in this paper: for instance, the events recognized by the grammar can be proposed to the shallow temporal extraction system, as the latter failed to recognize some of them in our evaluation. Additional work could also investigate the use of a meta-learning component to detect correct and incorrect information in either sub-system. In this paper, we have shown, however, that even the simple approach that we explored already produces competitive results that improve the performance of the whole system.

REFERENCES

Dorit ABUSCH (1994), Sequence of Tense Revisited: Two Semantic Accounts of Tense in Intensional Contexts, in Hans KAMPF, editor, *Ellipsis, Tense and Questions*, pp. 87–139, University of Amsterdam, DYANA deliverable R.2.2.B.

Peter ADOLPHS, Stephan OEPEN, Ulrich CALLMEIER, Berthold CRYSMANN, Dan FLICKINGER, and Bernd KIEFER (2008), Some Fine Points of Hybrid Natural Language Parsing, in *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Genoa, Italy.

Jan ALEXANDERSSON, Ralf ENGEL, Michael KIPP, Stephan KOCH, Uwe KÜSSNER, Norbert REITHINGER, and Manfred STEDE (2000), Modeling Negotiation Dialogs, in Wolfgang WAHLSTER, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 441–451, Springer-Verlag, Berlin Heidelberg New York, Artificial Intelligence edition.

James ALLEN (1983), Maintaining Knowledge about Temporal Intervals, *Communications of the ACM*, 26(11):832–843.

James ALLEN (1984), Towards a General Theory of Action and Time, *Artificial Intelligence*, 23:123–154.

Gabor ANGELI, Christopher D. MANNING, and Daniel JURAFSKY (2012), Parsing Time: Learning to Interpret Time Expressions, in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 446–455, Association for Computational Linguistics, Montréal, Canada.

Emily M. BENDER, Dan FLICKINGER, and Stephan OEPEN (2002), The Grammar Matrix: An Open-Source Starter-Kit for the Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars, in John CARROLL, Nelleke OOSTDIJK, and Richard SUTCLIFFE, editors, *Procedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pp. 8–14, Taipei, Taiwan.

Emily M. BENDER, Dan FLICKINGER, Stephan OEPEN, Annemarie WALSH, and Timothy BALDWIN (2004), Arboretum. Using a precision grammar for grammar checking in CALL, in *Proceedings of the InSTIL Symposium on NLP and Speech Technologies in Advanced Language Learning Systems*, Venice, Italy.

Steven BETHARD, James H. MARTIN, and Sara KLINGENSTEIN (2007), Finding Temporal Structure in Text: Machine Learning of Syntactic Temporal Relations, *International Journal of Semantic Computing*, 1(4):441–457.

André BITTAR, Pascal AMSILI, Pascal DENIS, and Laurence DANLOS (2011), French TimeBank: An ISO-TimeML Annotated Reference Corpus, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pp. 130–134, Association for Computational Linguistics, Portland, Oregon, USA.

Daniel G. BOBROW, Bob CHESLOW, Cleo CONDORAVDI, Lauri KARTTUNEN, Tracy Holloway KING, Rowan NAIRN, Valeria DE PAIVA, Charlotte PRICE, and Annie ZAENEN (2007), PARC's Bridge and Question Answering System, in Tracy Holloway KING and Emily M. BENDER, editors, *Proceedings of the GEAF07 Workshop*, pp. 46–66, CSLI Publications, Stanford, CA, `http://csli-publications.stanford.edu/GEAF/2007/geaf07-toc.html`.

Olivier BONAMI (2002), A syntax-semantics interface for tense and aspect in French, in Frank Van EYNDE, Lars HELLAN, and Dorothee BEERMANN, editors, *The Proceedings of the 8th International Conference on Head-Driven Phrase Structure Grammar*, pp. 31–50, CSLI Publications, Stanford.

Francis BOND, Stephan OEPEN, Melanie SIEGEL, Ann COPESTAKE, and Dan FLICKINGER (2005), Open Source Machine Translation with DELPH-IN, in *Proceedings of the Open-Source Machine Translation Workshop at the 10th Machine Translation Summit*, pp. 15–22, Phuket, Thailand.

Philip BRAMSEN, Pawan DESHPANDE, Yoong Keok LEE, and Regina BARZILAY (2006), Inducing Temporal Graphs, in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pp. 189–198, Sydney, Australia.

Bertram C. BRUCE (1972), A Model for Temporal References and Its Application in a Question Answering Program, *Artificial Intelligence*, 3(1–3):1–25.

Caroline BRUN (1998), Terminology Finite-State Preprocessing for Computational LFG, in *COLING '98 Proceedings of the 17th International Conference on Computational Linguistics*, volume 1, pp. 196–200, Association for Computational Linguistics, Stroudsburg, PA, USA.

Miriam BUTT, Tracy Holloway KING, María-Eugenia NIÑO, and Frédérique SEGOND (1999), *A Grammar Writer's Cookbook*, CSLI Publications, Stanford.

Ulrich CALLMEIER (2000), PET – A Platform for Experimentation with Efficient HPSG Processing Techniques, *Natural Language Engineering*, 6(1):99–108 (Special Issue on Efficient Processing with HPSG).

Tommaso Caselli, Valentina Bartalesi Lenzi, Rachele Sprugnoli, Emanuele Pianta, and Irina Prodanof (2011), Annotating Events, Temporal Expressions and Relations in Italian: the It-TimeML Experience for the Ita-TimeBank, in *Proceedings of the 5th Linguistic Annotation Workshop*, pp. 143–151, Association for Computational Linguistics, Portland, Oregon, USA, http://www.aclweb.org/anthology/W11-0418.

Nathanael Chambers and Daniel Jurafsky (2008), Jointly Combining Implicit Constraints Improves Temporal Ordering, in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 698–706, Association for Computational Linguistics, Honolulu, Hawaii.

Nathanael Chambers, Shan Wang, and Dan Jurafsky (2007), Classifying temporal relations between events, in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic.

Eugene Charniak and Mark Johnson (2005), Coarse-to-fine n-best parsing and MaxEnt discriminative reranking, in *Proceedings of the 43rd Annual Meeting of the ACL*, pp. 173–180, Association for Computational Linguistics, Ann Arbor.

Yuchang Cheng, Masayuki Asahara, and Yuji Matsumoto (2008), Constructing a Temporal Relation Tagged Corpus of Chinese Based on Dependency Structure Analysis, *Computational Linguistics and Chinese Language Processing*, 13(2):171–196.

Bernard Comrie (1986), Tense in Indirect Speech, *Folia Linguistica*, 20:265–296.

Ann Copestake (2002), *Implementing Typed Feature Structure Grammars*, CSLI Publications, Stanford.

Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard (2005), Minimal Recursion Semantics: An Introduction, *Journal of Research on Language and Computation*, 3(2-3):281–332.

Francisco Costa (2013), *Processing Temporal Information in Unstructured Documents*, Ph.D. thesis, Universidade de Lisboa, Lisbon, http://nlx.di.fc.ul.pt/~fcosta/papers/phdthesis.pdf.

Francisco Costa and António Branco (2010a), LXGram: A Deep Linguistic Processing Grammar for Portuguese, in *Lecture Notes in Artificial Intelligence*, volume 6001, pp. 86–89, Springer, Berlin, http://nlx.di.fc.ul.pt/~fcosta/papers/propor2010.pdf.

Francisco Costa and António Branco (2010b), Temporal Information Processing of a New Language: Fast Porting with Minimal Resources, in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL2010)*, pp. 671–677, Association for Computational Linguistics, Uppsala, Sweden, http://nlx.di.fc.ul.pt/~fcosta/papers/acl2010.pdf.

Francisco COSTA and António BRANCO (2012a), Backshift and Tense Decomposition, in Stefan MÜLLER, editor, *Proceedings of the 19th International Conference on Head-Driven Phrase Structure Grammar, Chungnam National University Daejeon*, pp. 86–106, ISSN 15351793, `http://nlx.di.fc.ul.pt/~fcosta/papers/hpsg2012.pdf`.

Francisco COSTA and António BRANCO (2012b), Extracting Temporal Information from Portuguese Texts, in Helena CASELI, Aline VILLAVICENCIO, António TEIXEIRA, and Fernando PERDIGÃO, editors, *Computational Processing of the Portuguese Language – 10th International Conference, PROPOR 2012*, volume 7243 of *Lecture Notes in Artificial Intelligence*, pp. 99–105, Springer, Berlin, Germany, `http://nlx.di.fc.ul.pt/~fcosta/papers/propor2012.pdf`.

Francisco COSTA and António BRANCO (2012c), LX-TimeAnalyzer: A Temporal Information Processing System for Portuguese, Technical Report DI-FCUL-TR-2012-01, Universidade de Lisboa, Faculdade de Ciências, Departamento de Informática, `http://nlx.di.fc.ul.pt/~fcosta/papers/tr2012-1.pdf`.

Francisco COSTA and António BRANCO (2012d), TimeBankPT: A TimeML Annotated Corpus of Portuguese, in Nicoletta CALZOLARI, Khalid CHOUKRI, Thierry DECLERCK, Mehmet Uğur DOĞAN, Bente MAEGAARD, Joseph MARIANI, Jan ODIJK, and Stelios PIPERIDIS, editors, *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, pp. 3727–3734, European Language Resources Association (ELRA), Istanbul, Turkey, ISBN 978-2-9517408-7-7, `http://nlx.di.fc.ul.pt/~fcosta/papers/lrec2012.pdf`.

Berthold CRYSMANN, Anette FRANK, Bernd KIEFER, Stefan MÜLLER, Günter NEUMANN, Jakub PISKORSKI, Ulrich SCHÄFER, Melanie SIEGEL, Hans USZKOREIT, Feiyu XU, Markus BECKER, and Hans-Ulrich KRIEGER (2002), An Integrated Architecture for Shallow and Deep Processing, in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL '02)*, pp. 441–448, Association for Computational Linguistics, Philadelphia, Pennsylvania.

James R. CURRAN, Stephen CLARK, and Johan BOS (2007), Linguistically Motivated Large-Scale NLP with C&C and Boxer, in *ACL '07 Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pp. 33–36, Association for Computational Linguistics, Stroudsburg, PA, USA.

Donald DAVIDSON (1967), The Logical Form of Action Sentences, in Nicholas RESCHER, editor, *The Logic of Decision and Action*, University of Pittsburgh Press.

Henriëtte DE SWART (1998), Aspect Shift and Coercion, *Natural Language and Linguistic Theory*, 16:347–385.

Henriëtte DE SWART (2000), Tense, aspect and coercion in a cross-linguistic perspective, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the Berkeley Formal Grammar Conference*, CSLI Publications, Stanford.

Rennat DECLERCK (1990), Sequence of Tenses in English, *Folia Linguistica*, 24:513–544.

Pascal DENIS and Philippe MULLER (2010), Comparison of different algebras for inducing the temporal structure of texts, in *Proceedings of COLING 2010*, pp. 250–258, Beijing, PRC.

Pascal DENIS and Philippe MULLER (2011), Predicting Globally-Coherent Temporal Structures from Texts via Endpoint Inference and Graph Decomposition, in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 2011*.

David R. DOWTY (1979), *Word Meaning and Montague Grammar: the Semantics of Verbs and Times in Generative Semantics and Montague's PTQ*, Reidel, Dordrecht.

Christiane FELLBAUM, editor (1998), *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA.

Lisa FERRO, Laurie GERBER, Inderjeet MANI, Beth SUNDHEIM, and George WILSON (2004), TIDES 2003 standard for the annotation of temporal expressions, Technical report, The MITRE Corporation, McLean, Virginia.

Dan FLICKINGER (2000), On Building a More Efficient Grammar by Exploiting Types, *Natural Language Engineering*, 6(1):15–28 (Special Issue on Efficient Processing with HPSG).

Dan FLICKINGER, Jan Tore LØNNING, Helge DYVIK, Stephan OEPEN, and Francis BOND (2005), SEM-I Rational MT: Enriching Deep Grammars with a Semantic Interface for Scalable Machine Translation, in *Proceedings of the 10th Machine Translation Summit*, pp. 165–172, Phuket, Thailand.

Maria FLOURAKI (2006), Constraining Aspectual Composition, in Stefan MÜLLER, editor, *The Proceedings of the 13th International Conference on Head-Driven Phrase Structure Grammar*, pp. 140–157, CSLI Publications, Stanford.

Corina FORĂSCU and Dan TUFIŞ (2012), Romanian TimeBank: An Annotated Parallel Corpus for Temporal Information, in Nicoletta CALZOLARI, Khalid CHOUKRI, Thierry DECLERCK, Mehmet Uğur DOĞAN, Bente MAEGAARD, Joseph MARIANI, Jan ODIJK, and Stelios PIPERIDIS, editors, *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, European Language Resources Association (ELRA), Istanbul, Turkey, ISBN 978-2-9517408-7-7.

Anette FRANK, Marcus BECKER, Berthold CRYSMANN, Bernd KIEFER, and Ulrich SCHÄFER (2003), Integrated Shallow and Deep Parsing: TopP meets HPSG, in *Proceedings of the 41st Annual Meeting of the Association for*

Computational Linguistics (ACL 2003), pp. 104–111, Association for Computational Linguistics, Sapporo, Japan.

David GOSS-GRUBBS (2005), *An Approach to Tense and Aspect in Minimal Recursion Semantics*, Master's thesis, University of Washington, Seattle, Washington.

Claire GROVER and Alex LASCARIDES (2001), XML-Based Data Preparation for Robust Deep Parsing, in *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 252–259, Toulouse, France.

Eun Young HA, Alok BAIKADI, Carlyle LICATA, and James C. LESTER (2010), NCSU: Modeling Temporal Relations with Markov Logic and Lexical Ontology, in Katrin ERK and Carlo STRAPPARAVA, editors, *SemEval 2010 – 5th International Workshop on Semantic Evaluation – Proceedings of the Workshop*, pp. 341–344, Uppsala University, Uppsala, Sweden.

Janet HITZEMAN, Marc MOENS, and Claire GROVER (1995), Algorithms for Analysing the Temporal Structure of Discourse, in *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 253–260, Dublin, Ireland.

Norbert HORNSTEIN (1991), *As Time Goes By*, MIT Press, Cambridge, USA.

Seohyun IM, Hyunjo YOU, Hayun JANG, Seungho NAM, and Hyopil SHIN (2009), KTimeML: Specification of Temporal and Event Expressions in Korean Text, in *Proceedings of the 7th Workshop on Asian Language Resources*, pp. 115–122, Association for Computational Linguistics, Stroudsburg, PA, USA.

George H. JOHN and Pat LANGLEY (1995), Estimating Continuous Distributions in Bayesian Classifiers, in *Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, San Mateo.

Hans KAMP and Uwe REYLE (1993), *From Discourse to Logic: An Introduction to Modeltheoretic Semantics, Formal Logic and Discourse Representation Theory*, Kluwer, Dordrecht.

Ronald M. KAPLAN and Joan BRESNAN (1982), Lexical-Functional Grammar: A Formal System for Grammatical Representation, in Joan BRESNAN, editor, *The Mental Representation of Grammatical Relations*, MIT Press Series on Cognitive Theory and Mental Representation, chapter 4, pp. 173–281, MIT Press, Cambridge, Massachusetts.

Ronald M. KAPLAN, John T. MAXWELL III, Tracy Holloway KING, and Richard CROUCH (2004), Integrating Finite-state Technology with Deep LFG Grammars, in *Proceedings of the Workshop on Combining Shallow and Deep Processing for NLP (ESSLLI 2004)*, Nancy, France.

Oleksandr KOLOMIYETS, Steven BETHARD, and Marie-Francine MOENS (2011), Model-Portability Experiments for Textual Temporal Analysis, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human*

Language Technologies, pp. 271–276, Association for Computational Linguistics, Portland, Oregon, USA, `http://www.aclweb.org/anthology/P11-2047`.

Hans-Ulrich KRIEGER and Ulrich SCHÄFER (1994), TDL – A Type Description Language for Constraint-Based Grammars, in *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 893–899, Kyoto, Japan.

John LAFFERTY, Andrew MCCALLUM, and Fernando C. N. PEREIRA (2001), Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, in *Proceedings of the 8th International Conference on Machine Learning 2001 (ICML 2001)*, pp. 282–289.

Alex LASCARIDES and Nicholas ASHER (1993), Temporal Interpretation, Discourse Relations, and Common Sense Entailment, *Linguistics and Philosophy*, 16:437–493.

Chong Min LEE (2010), Temporal Relation Identification with Endpoints, in *HLT-SRWS '10 Proceedings of the NAACL HLT 2010 Student Research Workshop*, pp. 40–45, Association for Computational Linguistics, Stroudsburg, PA, USA.

Xiao LING and Daniel S. WELD (2010), Temporal Information Extraction, in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*.

Hector LLORENS, Leon DERCZYNSKI, Robert GAIZAUSKAS, and Estela SAQUETE (2012), TIMEN: An Open Temporal Expression Normalization Resource, in Nicoletta CALZOLARI, Khalid CHOUKRI, Thierry DECLERCK, Mehmet Uğur DOĞAN, Bente MAEGAARD, Joseph MARIANI, Jan ODIJK, and Stelios PIPERIDIS, editors, *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, pp. 3044–3051, European Language Resources Association (ELRA), Istanbul, Turkey.

Hector LLORENS, Estela SAQUETE, and Borja NAVARRO (2010a), TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2, in Katrin ERK and Carlo STRAPPARAVA, editors, *SemEval 2010 – 5th International Workshop on Semantic Evaluation – Proceedings of the Workshop*, pp. 284–291, Uppsala University, Uppsala, Sweden.

Hector LLORENS, Estela SAQUETE, and Borja NAVARRO-COLORADO (2010b), TimeML Events Recognition and Classification: Learning CRF Models with Semantic Roles, in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 725–733, Coling 2010 Organizing Committee, Beijing, PRC.

Paweł MAZUR and Robert DALE (2010), WikiWars: A New Corpus for Research on Temporal Expressions, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'10)*, pp. 913–922.

Nurit MELNIK (2005), From "Hand-Written" to Computationally Implemented HPSG Theories, in Stefan MÜLLER, editor, *The Proceedings of the 12th*

International Conference on Head-Driven Phrase Structure Grammar, pp. 311–321, CSLI Publications, Stanford.

Laura MICHAELIS (2006), Tense in English, in Bas AARTS and April MCMAHON, editors, *The Handbook of English Linguistics*, Blackwell, Oxford.

Laura A. MICHAELIS (2011), Stative by Construction, *Linguistics*, 49:1359–1400.

Marc MOENS and Mark STEEDMAN (1988), Temporal ontology and temporal reference, *Computational Linguistics*, 14(2):15–28.

MUC-6 (1995), *Proceedings of the Sixth Message Understanging Conference (MUC-6)*, Defense Advanced Research Projects Agency.

MUC-7 (1998), *Proceedings of the Seventh Message Understanging Conference (MUC-7)*, Defense Advanced Research Projects Agency.

Stefan MÜLLER and Walter KASPER (2000), HPSG Analysis of German, in Wolfgang WAHLSTER, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 238–253, Springer-Verlag, Berlin Heidelberg New York, Artificial Intelligence edition.

Matteo NEGRI and Luca MARSEGLIA (2004), Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004, Technical report, Trento.

Eric NICHOLS, Francis BOND, Darren SCOTT APPLING, and Yuji MATSUMOTO (2007), Combining Resources for Open Source Machine Translation, in *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*, pp. 1–10, Skövde, Sweden.

Eric NICHOLS, Francis BOND, Takaaki TANAKA, Sanae FUJITA, and Daniel FLICKINGER (2006), Robust Ontology Acquisition from Multiple Sources, in *Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pp. 10–17, Sydney, Australia, `http://www.aclweb.org/anthology/W/W06/W06-0502`.

Lars NYGAARD, Jan Tore LØNNING, Torbjørn NORDGÅRD, and Stephan OEPEN (2006), Using a Bi-Lingual Dictionary in Lexical Transfer, in *Proceedings of the 11th Conference of the European Association for Machine Translation*, Oslo, Norway.

Stephan OEPEN, Kristina TOUTANOVA, Stuart SHIEBER, Christopher MANNING, Dan FLICKINGER, and Thorsten BRANTS (2002), The LinGO Redwoods Treebank: Motivation and Preliminary Applications, in *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pp. 1253–1257, Taipei, Taiwan.

Barbara PARTEE (1973), Some Structural Analogies Between Tenses and Pronouns in English, *The Journal of Philosophy*, 70:601–609.

Carl POLLARD and Ivan SAG (1994), *Head-Driven Phrase Structure Grammar*, Chicago University Press and CSLI Publications, Stanford.

Georgiana PUŞCAŞU (2007), WVALI: Temporal Relation Identification by Syntactico-Semantic Analysis, in *Proceedings of SemEval-2007*, pp. 484–487, Association for Computational Linguistics, Prague, Czech Republic.

James PUSTEJOVSKY (1991), The Syntax of Event Structure, *Cognition*, 41:47–81.

James PUSTEJOVSKY, José CASTAÑO, Robert INGRIA, Roser SAURÍ, Robert GAIZAUSKAS, Andrea SETZER, and Graham KATZ (2003a), TimeML: Robust Specification of Event and Temporal Expressions in Text, in *IWCS-5, Fifth International Workshop on Computational Semantics*.

James PUSTEJOVSKY, Patrick HANKS, Roser SAURÍ, Andrew SEE, Robert GAIZAUSKAS, Andrea SETZER, Dragomir RADEV, Beth SUNDHEIM, David DAY, Lisa FERRO, and Marcia LAZO (2003b), The TIMEBANK Corpus, in *Proceedings of Corpus Linguistics 2003*, pp. 647–656.

James PUSTEJOVSKY, Jessica LITTMAN, Roser SAURÍ, and Marc VERHAGEN (2006), TimeBank 1.2 Documentation, `http://timeml.org/site/timebank/documentation-1.2.html`.

James PUSTEJOVSKY and Amber STUBBS (2011), Increasing Informativeness in Temporal Annotation, in *Proceedings of the Fifth Law Workshop (LAW V)*, pp. 152–160, Association for Computational Linguistics, Portland, Oregon, USA.

Hans REICHENBACH (1947), *Elements of Symbolic Logic*, University of California Press, Berkeley.

Matthew RICHARDSON and Pedro DOMINGOS (2006), Markov Logic Networks, *Machine Learning*, 62(1):107–136.

Joshua P. RODRÍGUEZ (2004), *Interpreting the Spanish* Imperfecto*: Issues of Aspect, Modality, Tense, and Sequence of Tense*, Ph.D. thesis, The Ohio State University, Columbus, Ohio.

C.J. RUPP, Jörg SPILKER, Martin KLARNER, and Karsten L. WORM (2000), Combining Analyses from Various Parsers, in *Verbmobil: Foundations of Speech-to-Speech Translation*, pp. 311–320, Springer, Berlin, Artificial Intelligence edition.

Ivan A. SAG, Thomas WASOW, and Emily M. BENDER (2003), *Syntactic Theory – A Formal Introduction*, CSLI Publications, Stanford.

Roser SAURÍ, Jessica LITTMAN, Bob KNIPPEN, Robert GAIZAUSKAS, Andrea SETZER, and James PUSTEJOVSKY (2006), TimeML Annotation Guidelines: Version 1.2.1, `http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf`, manuscript (retrieved October 19, 2012).

Ulrich SCHÄFER (2006), *Integrating Deep and Shallow Natural Language Processing Components – Representations and Hybrid Architectures*, Ph.D. thesis, Unversität des Saarlandes, Saarbrücken.

João SILVA, António BRANCO, Sérgio CASTRO, and Francisco COSTA (2012), Deep, Consistent and also Useful: Extracting Vistas from Deep Corpora for Shallower Tasks, in Jan HAJIČ, Koenraad De SMEDT, Marko TADIĆ, and António BRANCO, editors, *Proceedings of the Workshop on Advanced Treebanking at the 8th International Conference on Language Resources and Evaluation (LREC'12)*, pp. 45–52, Istanbul, Turkey.

Manfred STEDE, Stefan HAAS, and Uwe KÜSSNER (1998), Understanding and tracking temporal descriptions in dialogue, in Bernhard SCHRÖDER, Winfried LENDERS, Wolfgang HESS, and Thomas PORTELE, editors, *Computers, Linguistics, and Phonetics between Language and Speech – Proceedings of the 4th Conference on Natural Language Processing KONVENS'98*, Peter Lang, Frankfurt, Germany.

Jannik STRÖTGEN and Michael GERTZ (2013), Multilingual and Cross-domain Temporal Tagging, *Language Resources and Evaluation*, 47(2):269–298.

Kristina TOUTANOVA, Christopher D. MANNING, Dan FLICKINGER, and Stephan OEPEN (2005), Stochastic HPSG Parse Selection using the Redwoods Corpus, *Journal of Research on Language and Computation*, 3(1):83–105.

Naushad UZZAMAN and James ALLEN (2011), Temporal Evaluation, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 351–356, Association for Computational Linguistics, Portland, Oregon, USA.

Naushad UZZAMAN and James F. ALLEN (2010), TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text, in Katrin ERK and Carlo STRAPPARAVA, editors, *SemEval 2010 – 5th International Workshop on Semantic Evaluation – Proceedings of the Workshop*, pp. 276–283, Uppsala University, Uppsala, Sweden.

Naushad UZZAMAN, Hector LLORENS, Leon DERCZYNSKI, Marc VERHAGEN, James ALLEN, and James PUSTEJOVSKY (2013), SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations, in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*.

Frank VAN EYNDE (1998), Tense, Aspect and Negation, in Frank VAN EYNDE and Paul SCHMIDT, editors, *Linguistic Specifications for Typed Feature Structure Formalisms. Studies in machine Translation and Natural language Processing*, volume 10, pp. 209–280, Luxembourg.

Frank VAN EYNDE (2000), A constraint-based semantics for tenses and temporal auxiliaries, in Ronnie CANN, Claire GROVER, and Philip MILLER, editors, *Grammatical interfaces in HPSG*, pp. 231–249, CSLI Publications, Stanford University.

Eric VELLDAL (2007), *Empirical Realization Ranking*, Ph.D. thesis, University of Oslo, Oslo.

Zeno VENDLER (1967), Verbs and Times, in *Linguistics in Philosophy*, pp. 97–121, Cornell University Press, Ithaca, New York.

Marc VERHAGEN, Robert GAIZAUSKAS, Frank SCHILDER, Mark HEPPLE, Jessica MOSZKOWICZ, and James PUSTEJOVSKY (2009), The TempEval challenge: identifying temporal relations in text, *Language Resources and Evaluation*, 43(2):161–179, Special Issue: Computational Semantic Analysis of Language: SemEval-2007 and Beyond.

Marc VERHAGEN, Robert GAIZAUSKAS, Frank SCHILDER, Mark HEPPLE, and James PUSTEJOVSKY (2007), SemEval-2007 Task 15: TempEval temporal relation identification, in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 75–80, Association for Computational Linguistics, Prague, Czech Republic.

Marc VERHAGEN and James PUSTEJOVSKY (2008), Temporal Processing with the TARSQI Toolkit, in *COLING 2008: Companion Volume: Demonstrations*, pp. 189–192.

Marc VERHAGEN, Roser SAURÍ, Tommaso CASELLI, and James PUSTEJOVSKY (2010), SemEval-2010 Task 13: TempEval-2, in Katrin ERK and Carlo STRAPPARAVA, editors, *SemEval 2010 – 5th International Workshop on Semantic Evaluation – Proceedings of the Workshop*, pp. 51–62, Uppsala University, Uppsala, Sweden.

Wolfgang WAHLSTER, editor (2000), *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer, Berlin, Artificial Intelligence edition.

Nianwen XUE and Yuping ZHOU (2010), Applying Syntactic, Semantic and Discourse Constraints to Chinese Temporal Annotation, in *Proceedings of COLING 2010*, pp. 1363–1372, Beijing, PRC.

Katsumasa YOSHIKAWA, Sebastian RIEDEL, Masayuki ASAHARA, and Yuji MATSUMOTO (2009), Jointly Identifying Temporal Relations with Markov Logic, in *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*.

Kei YOSHIMOTO and Yoshiki MORI (2002), A compositional Semantics for Complex Tenses in Japanese, in Frank VAN EYNDE, Lars HELLAN, and Dorothee BEERMANN, editors, *The Proceedings of the 8th International Conference on Head-Driven Phrase Structure Grammar*, pp. 300–319, CSLI Publications, Stanford, `http://cslipublications.stanford.edu/HPSG/2/`.

Xujian ZHAO, Peiquan JIN, and Lihua YUE (2010), Automatic Temporal Expression Normalization with Reference Time Dynamic-Choosing, in *COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 1498–1506, Association for Computational Linguistics, Stroudsburg, PA, USA.

Yuping ZHOU and Nianwen XUE (2011), Discourse-constrained Temporal Annotation, in *Proceedings of the 5th Linguistic Annotation Workshop (LAW V '11)*, pp. 161–169, Association for Computational Linguistics, Stroudsburg, PA, USA.

This work is licensed under the Creative Commons Attribution 3.0 Unported License.
http://creativecommons.org/licenses/by/3.0/

An informal discovery procedure for two-level rules

Kimmo Koskenniemi
Department of Modern Languages, University of Helsinki, Finland

ABSTRACT

The paper shows how a certain kind of underlying representations (or deep forms) of words can be constructed in a straightforward manner through aligning the surface forms of the morphs of the word forms. The inventory of morphophonemes follows directly from this alignment. Furthermore, the two-level rules which govern the different realisations of such morphophonemes follow fairly directly from the previous steps. The alignment and rules are based upon an approximate general metric among phonemes, e.g., articulatory features, that determines which alternations are likely or possible. This enables us to summarise contexts for the different realisations.

Keywords: morphological analysis, discovery procedure, two-level rules, two-level morphology, IPA

1 INTRODUCTION

The orientation of this paper is linguistic rather than statistical, and the general framework is not taken from machine learning. The aim of the procedure that this work details is to assist rather than to replace the linguist. The scheme makes use of the common knowledge that human linguists have. The procedure is intended to make a part of such knowledge operational. In order to use the procedure, the linguist must select examples which contain only regular (morpho)phonological alternations. The alternations must be of the types for which the procedure has general models, e.g., assimilations, agreements or phonotactic constraints. A good choice of examples is essential for getting good and general rules.

Linguists have a special talent to cope with *regularities* and *exceptions*. A human linguist is able to consider factors beyond plain frequencies and, e.g., recognise the fact that certain morphophonological phenomena are closed, and no new words will follow them, and that some others are productive and readily extend to new words. Using such knowledge, the linguist can sometimes make sense of phenomena which might remain fuzzy for present machine learning and statistical methods.

The aims of the procedure sketched here are (1) to partition a set of example word forms into stems and inflectional morphemes by aligning (character by character) the stems and inflectional morphs, (2) to establish morphophonemic representations for the stems and affixes, and (3) to deduce a set of two-level rules which express the general context conditions (according to which the morphophonemes are realised in the examples and in any similar words). Through such steps, lexicon representations and rules for an inflectional class can be established. Applying the procedure to all productive inflectional classes is needed in order to describe the morphology of a language.

General (but approximate) linguistic knowledge about phonology guides the mechanical procedure presented in this paper. In particular, knowledge of the kinds of phonological alternations that are common in the languages of the world, and the kinds of phonological contexts that such alternations typically occur in, are used. Possible alternations can be, e.g., *assimilations* (where adjacent sounds become more similar to each other), *dissimilations* (where similar sounds become more distinct from each other), *metathesis* (where two sounds are swapped), *phonotactic constraints* (where, e.g., a certain type of syllable structure is enforced), agreements or *harmonies* (where, e.g., some vowels in the affixes become more similar to those in the word root).

Two-level morphology is used here because, in that framework, individual rules can be kept quite *independent* of each other.[1] The original form of the two-level morphology (Koskenniemi, 1983) is mathematically simple because each rule written by the linguist is a con-

[1] See, e.g., Karttunen (1993) for an overall introduction to two-level morphology.

straint which must be satisfied separately. On the other hand, this simplicity makes it more difficult to simulate generative phonology using two-level rules in cases where some phonemes can be affected by several unrelated rules.

A more complex form of the two-level formalism which uses rule conflict resolution mechanisms is more suitable for using similar underlying representations as the generative phonology. Rule conflicts can be resolved by pre-processing the individual rules using the whole grammar and by copying parts of rules into some other rules which makes the rules more complex, see Karttunen *et al.* (1987) for that approach.

The present approach uses *reduced versions* of the two-level grammars which neither use any conflict resolution mechanisms nor need them. In two-level grammars, right-arrow conflicts only arise when the same correspondence occurs in separate rules. This does not happen in the proposed procedure which produces a single rule for each realisation of a morphophoneme (even if such a rule may have several context parts). Left-arrow conflicts arise when different realisations of a morphophoneme have overlapping contexts. The conflict resolution mechanism of two-level compilers recognises the special case where one context is a proper subset of another, and resolves it by prioritising the smaller context. The grammars produced by the proposed procedure avoid left-arrow conflicts because the corresponding morphophonemes will be distinct from each other.

Rewrite rules must usually be applied in a specific order. Applying a rule changes the string, and the remaining rules depend on the earlier ones. The number of different orderings grows according to the factorial of the number of rules, e.g., five rules could be applied in 120 distinct orders but 20 rules in as many as 2,432,902,008,176,640,000. Phonological grammars using rewrite rules need a rule ordering, even if not all rules are equally sensitive to such. Discovering complex rewrite rule grammars is probably difficult because one must discover both the rules and their ordering. The reduced two-level grammars, on the other hand, avoid the rule ordering altogether because all rules are applied in parallel.

The procedure presented here treats the underlying forms of morphemes and the morphophonemes in an extremely concrete manner. *Morphophonemes* are represented just as the *combinations of the corre-*

sponding letters (or phonemes) which we can observe in the surface forms. On the one hand, such an interpretation of morphophonemes is crude, but on the other hand, it is a fact that anybody can observe. *Lexical representations* are sequences of *lexical characters*, each of which is either a letter or a morphophoneme.

Using a systematically selected set of surface forms, the procedure creates lexicon entries and rules for such words. The entries contain morphophonemes deduced by the procedure. The deduced rules are simple two-level rules. The resulting rules can be compiled using the existing two-level compilers, such as the open source HFST-TWOLC (Silfverberg and Lindén, 2009) or the proprietary Xerox LEXC.[2] This paper describes the plan for the actual procedure, including some feasibility estimates according to which the approach appears to be tractable. Unfortunately, an implementation is not yet available (but cf. Section 14).

This work differs from unsupervised discovery of morphology where the lexical units and rules would be induced from raw corpus data. The approach is based on the observation that many linguists find it difficult to express their intuitions as formal rules, whereas they are comfortable with providing concrete examples. This work makes use of this kind of human supervision. When implemented, the procedure would be a useful tool for a linguist. The informal procedure as discussed below, may even guide the linguist in designing rules and grammars even in the absence of an implementation. Theoretically oriented linguists might also discuss the merits and the shortcomings of the very concrete and objective interpretation for morphophonemes presented here.

This paper argues for the utility of phoneme by phoneme (or character by character) alignment of word forms and discusses its tractability. In particular, it presents a way how both the stem parts and the affix parts should and can be aligned. The alignment appears to be possible for a wide array of different types of languages. Such an alignment can be directly utilised in the two-level framework (but less so in the rewriting framework). Once a proper alignment is determined,

---

[2] See `www.stanford.edu/~laurik/.book2software/twolc.pdf` and `http://www.cis.upenn.edu/~cis639/docs/twolc.html` for the documentation of the two-level grammar formalism.

the lexical representations of stems and affixes are algorithmically and uniquely determined.

It is argued that, in this framework, the deduction of the necessary morphophonological two-level rules is easier than using the rewrite rule framework. In particular, it is shown that the *deduction of rules for one morphophoneme is entirely independent* of the rules for other morphophonemes. In principle, one may discover rules for various kinds of morphophonemic alternations provided that the alternations are (morpho)phonological in their nature, and do not involve suppletion or otherwise introduce, delete or change larger units. Specific procedures for each type of conditioning are needed, but can be designed.

The main claim of this paper is that the framework presented has a better potential to succeed in discovering regularities in languages which have an elaborate morphophonology. Other approaches appear to perform best when applied to English or other languages with fairly simple morphology. In particular, as compared to the machine learning oriented approaches, the framework proposed here appears to be able to cope with more demanding phenomena such as interdigitation of Semitic languages, phenomena based on syllable structures, agreements, metathesis of non-contiguous segments, and the like. This paper argues for the validity of this claim. Deeper understanding can be reached when an implementation is available and has been applied to a number of different languages.

2 PAST WORK

Most of the recent work on the theme of morpho(phono)logical discovery methods has been done within the framework of unsupervised machine learning. These start from large unannotated corpora and other raw language resources and they try to describe the inflection of words in an economic manner without human intervention. The present paper operates differently, and it aims to discover more precise rules than what the unsupervised approaches can.

Gildea and Jurafsky (1995) report experiments with discovering finite-state transducers from large sets of examples. They extend and improve the OSTIA algorithm (Oncina *et al.*, 1993) by aligning the examples using an edit distance based on the phonological binary features of phonemes and use training data of tens of thousands of words in the general framework of Mitchell (1982).

Theron and Cloete (1997) studied a problem where the starting point was a list of pairs consisting of an inflected word form and the corresponding base form. Their work was based on edit distances when aligning the inflected and the base form with each other. Their algorithm deduced the context parts of the two-level rules by starting from the full lists of correct contexts and then by truncating them as long as the rules still accounted for the correct forms.

Yarowsky and Wicentowski (2000) present a method for deducing stem alternation rules and morphological analysis applicable for languages similar to the Indo-European languages which are nowadays spoken in Europe. The method makes use of initial tables of endings, an unannotated corpus and a collection of candidate noun, adjective and verb roots. Roots of word forms and rules are deduced using frequency statistics. The method of identifying roots is trained and combines several models. Most of the irregular English words were learnt by the procedure.

Linguistica algorithm (Goldsmith, 2006) and Morfessor (Creutz and Lagus, 2004) represent word forms using sets of substrings and they utilise criteria such as the minimum description length (MDL). The end results of such processes are sets of strings which are similar to linguistic morphs (but not always the same). Concatenations of such sets model the word forms of the language. Goldwater and Johnson (2004) build on top of Goldsmiths Linguistica and reduce the sets further by introducing phonological rules.

The PhD dissertation of Chan (2008) on the induction of morphology and lexical categories includes a fairly comprehensive survey on previous work on machine learning of morphology. A few studies have a more linguistic conception of the phonological rules to be found. They usually assume that the procedure has access both to the underlying and the surface form of example words.

Johnson (1984) presented a discovery procedure for ordered rewrite rules in the framework of generative phonology. He starts from a given table of forms of a set of lexemes and assumes that the morphemes have been segmented, corresponding phonemes identified, and that the phonemes are represented using their distinctive features. Johnson claimed that his procedure can cope with the rule ordering by considering the contexts of the rules. The last rules in the cascade have contexts which are apparent in the surface forms, whereas the rules early in the sequence tend to use contexts which are less visible in the surface forms. The data on which the method was tested involved six rules for Japanese. He notes that the rules, underlying representations, and the rule orderings are not strongly determined by the data. Lots of computation was required and it resulted in multiple solutions that consisted of different rule orderings and different underlying representations. The goals of Johnson were otherwise similar to those of the present paper.

Touretzky *et al.* (1990) study rules needed in order to generate phonetic realisations out of underlying (morpho)phonemic representations. The rules are learnt step by step from input examples where both the underlying and surface forms are given. The learning occurs by approximating the contexts both from specific examples (which may be too narrow) and generalisations (which may be too broad) as context conditions.

Oflazer and Nirenburg (1999) and Oflazer *et al.* (2001) present a method for bootstrapping morphological analysers by combining human elicitation and machine learning. Human informants provide the examples used by the machine learning process to deduce rewrite rules necessary for accounting for the data. The method has been applied, e.g., to Polish, where alternations both in the stems and in the endings were captured. This interactive approach is relevant for the procedure presented in this paper, and maybe the best parts of these two could be combined in future.

A recent work by Hulden *et al.* (2011) studies the learning of dialectal morphologies using parallel corpora. The authors end up using parallel rewrite rules for describing the mapping between the standard Basque and one of its dialects. They use the FOMA rewrite rule system (Hulden, 2009) which is an open source replacement for Xerox XFST (Beesley and Karttunen, 2003).[3]

The present paper elaborates an earlier work, see Koskenniemi (1991) where a version of the procedure was sketched. The general

[3] Both FOMA and XFST support certain forms of parallel rewrite rules which avoid the rule ordering problems. On the other hand, such parallel replace rules have to be compiled together as a single unit which may become computationally heavy if there are many rules.

idea of first aligning and then deducing was presented there. The present paper formalises the alignment in order to make the potentially intractable task feasible. In addition, a distance metric is presented so that choosing among alternative solutions is better defined. Furthermore, this paper tells more explicitly how languages with infixation, prefixation, etc., can be handled.

3 SIMPLIFIED TWO-LEVEL MORPHOPHONOLOGY

Koskenniemi (1983, 1984) detail the origins of the two-level morphology, Karttunen (1993) presents a brief introduction, and Karttunen and Beesley (2001) presents a history of the two-level formalism.

In all versions of the two-level formalism, the *morphophonological rules* are based on just two representations of word forms: the *lexical representation* and the *surface representation*. There are no intermediate representations between these two. The lexical representation is a string of characters and serves as the underlying representation of *morphemes*.

In the present simplified formalism, the lexical characters of the morphemes may be:

- *phonemes* (or letters) which, in our simplified formalism, always correspond to themselves in the surface representation;
- *morphophonemes* which correspond to two or more alternative phonemes (or letters) in the surface representation; and
- *auxiliary symbols* which always correspond to zero in the surface representation and may be used either as boundary symbols between morphemes or as markers of grammatical categories.

We denote morphophonemes by the alternative letters they represent, e.g., **aä** represents a morphophoneme which can be realised either as **a** or **ä**.[4] Lexical representations are given here with spaces between the characters in order to make the morphophonemes explicit, e.g., **t a l o s s aä**. Surface representations are strings of letters and zeroes (**0**).

[4] Mathematically, these kinds of morphophonemes are tuples rather than sets. In the general case, the morphophoneme is a sequence of the surface letters that the morphophoneme is realised as. Thus, a morphophoneme for an **e** in the base form and alternating with an **i** in its inflected forms is different from an **i** in the base form alternating with an **e**. The shorter notation is used simply for brevity.

Zeroes act as place holders for morphophonemes which are deleted from surface forms. Within the two-level rules, zeroes are, however, treated as characters (rather than epsilons or null strings).

Let us consider a simplified example taken from the Finnish inflection of nouns. Word forms are constructed by affixing endings after the stem, e.g., **piha a** 'yard' + 'partitive', **piho i lla** 'yard' + 'plural' + 'at' and **piho j a** 'yard' + 'plural' + 'partitive'. A linguist would notice that the stem-final vowel is either **a** or **o** in the surface forms, and therefore we interpret it as a morphophoneme **ao** in the lexical form. Similarly, we notice that the plural morph is either **i** or **j** on the surface. Thus, we have the lexical representation **p i h ao** for 'yard' and **ij** for 'plural'. Now we can represent the correspondence of the (somewhat simplified) lexical and surface forms (where **ao** and **ij** are single and indivisible symbols):

```
p i h ao a     p i h ao ij l l a      p i h ao ij a
p i h a  a     p i h o  i  l l a      p i h o  j  a
```

Two-level rules specify how lexical and surface representations may correspond to each other. We need one rule for **ao**, and another for **ij**. Studying these and other examples, the linguist would notice that **ao:o** occurs before the plural morpheme **ij** and that the plural morpheme itself realises as **ij:j** if and only if it ends up between vowels on the surface. A linguist might write the two-level grammar as:

```
Alphabet a d e f g h i j k l m n o p r s t u v y ä ö
         ao:a ij:i ;
Vowel = a e i o u y ä ö ;
Rules
"ao in plural"      ao:o <=> _ ij: ;
"ij between vowels" ij:j <=> :Vowel _ :Vowel ;
```

The alphabet lists all letters and morphophonemes. In addition, it gives the default correspondences of the morphophonemes, here **ao:a** and **ij:i**. The first rule tells two things: first (the right arrow component), that **ao:o** can occur only if it is immediately followed by a lexical **ij**, and secondly (the left arrow component), that when a lexical **ao** occurs in this context, **o** is the only possible corresponding surface character. Similarly, the second rule says that **ij:j** may occur only between surface vowels and that in such a place **ij:i** is forbidden. The pair before the double arrow is called the *centre* of the rule. The *context part* is (or possibly several of such are) on the right of the double arrow. A context part consists of the *left context* and a *right context* separated from each other by an underscore. Two-level grammars usually consist of double arrow <=> rules which combine the requirements of right arrow => and left arrow <= rules.

Two-level rules are usually compiled into finite-state transducers (FSTs). A compiled rule FST accepts all correct examples, i.e. strings of character pairs like: **p:p i:i h:h ao:a** or **p:p i:i h:h ao:o ij:i l:l l:l a:a**. According to a common practice, we represent pairs with identical components with a single character: **p i h ao:a** and **p i h ao:o ij:i l l a**.

The linguist collects a set of examples before even starting to design any rules. The examples specify the task for the rules. This specification is in a form that can be communicated to and understood by other linguists. Furthermore, the generated two-level rules can be checked against these examples in order to verify their correctness – as described in Section 13.

The rules the linguist writes depend on the lexical representations one chooses. The experience of the author, when supervising linguists and students, indicates that when the lexical representations have been carefully designed, the writing of the two-level rules is easy and straightforward. Using a sufficiently large set of morphophonemes, one can keep the rules simple and independent of each other.

4 CHOOSING A PARADIGM OF SELECTED LEXEMES AND FORMS

For the discovery procedure, we collect a table of word forms, i.e. surface forms of inflected words. On each row of the table, we have the same lexeme (in different forms) and on each column, we have the same form (of different lexemes). To be more precise, on a row, there are (possibly slightly different) stems of the same lexeme. One must not include lexemes with *suppletion*, i.e. words where stems represent different lexical units, such as **good** and **bett(-er)**. It is not reasonable even to attempt to model suppletion using two-level rules (whereas rewrite rules can freely be used for such). Two-level rules should be used only for *natural* morphophonological alternations.

In order to describe the steps of the discovery procedure, we use an example given in Table 1, where six Finnish nouns are inflected in five different forms: singular nominative, partitive, and inessive, and plural partitive and inessive (the columns).

	SgNom	SgPtv	SgIne	PlPtv	PlIne
'house'	talo	taloa	talossa	taloja	taloissa
'crack'	särö	säröä	särössä	säröjä	säröissä
'cross'	risti	ristiä	ristissä	ristejä	risteissä
'notch'	lovi	lovea	lovessa	lovia	lovissa
'fish'	kala	kalaa	kalassa	kaloja	kaloissa
'dog'	koira	koiraa	koirassa	koiria	koirissa

Table 1: Example raw word forms

5 LENGTHS OF THE STEMS AND AFFIXES

Alignment consists of augmenting the word forms with zeroes where necessary, and inserting boundary symbols which separate the stems from the affixes. Zeroes are sometimes needed in order to make the surface forms of the corresponding stems and affixes equal in length. After this step, the stems which belong to the same lexeme should have the same number of characters (letters plus possible zeroes), and the same goes for affixes which belong to the same grammatical form.

After adding some zeroes, the lengths of the word forms in the table can be expressed as a sum of the length of the (constant length) stems of the lexeme m_i and the (constant length) affix parts of the grammatical form n_j. Initially, the procedure has the lengths of the raw forms as given in Table 2.

Next, the procedure decomposes the raw lengths of the word forms into a sum of the lengths of the stem and affixes. It may arrive at several solutions and one of these should outperform the other ones during the subsequent steps. The procedure tentatively assumes that the stems are (at least) as long as the singular nominative forms. Then, the affixes are at least as long as the difference between the inflected and nominative forms. In this way, the procedure arrives at the decomposition of lengths in Table 3. The decomposition is accurate in all other places except for the plural forms of **lovi** and **koira** (which

Table 2: Lengths of the raw word forms

	SgNom	SgPtv	SgIne	PlPtv	PlIne
talo	4	5	7	6	8
särö	4	5	7	6	8
risti	5	6	8	7	9
lovi	4	5	7	5	7
kala	4	5	7	6	8
koira	5	6	8	6	8

Table 3: Lengths of the raw word forms tentatively decomposed into the lengths of the stems and affixes

stem	SgNom 0	SgPtv 1	SgIne 3	PlPtv 2	PlIne 4
talo 4	4+0=4	4+1=5	4+3=7	4+2=6	4+4=8
särö 4	4+0=4	4+1=5	4+3=7	4+2=6	4+4=8
risti 5	5+0=5	5+1=6	5+3=8	5+2=7	5+4=9
lovi 4	4+0=4	4+1=5	4+3=7	4+2=6>5	4+4=8>7
kala 4	4+0=4	4+1=5	4+3=7	4+2=6	4+4=8
koira 5	5+0=5	5+1=6	5+3=8	5+2=7>6	5+4=9>8

are one character too short). Thus, we insert a zero character into those word forms. The zero could be inserted anywhere in the word form and the procedure must usually evaluate several possibilities.

Furthermore, one must allow for more zeroes than the minimum amount to be added if there are more substantial (but regular) alternations within the stem. The calculation gives one or more hypotheses for the lengths of stems and affixes. The procedure proceeds first with the above assumption for the lengths and backtracks only if necessary.

The procedure must also be prepared to consider alternative partitions. One could, e.g., have shorter stems and longer affixes. Not too many alternatives exist, and the procedure can enumerate the decompositions of the lengths without problems and choose the best (or the only possible) alternative during the next steps of the procedure.

Up to now, the procedure has made no assumptions about the position of the stems. They might be at the beginning, end, or somewhere in the middle of the word forms. The stem might even be noncontiguous, i.e. interrupted by inflectional parts. One could claim that establishing the (tentative) lengths of morphemes as the first step is not necessary. It could be solved later, or as a part of the whole task using, e.g., dynamic programming. Such computation, however, appears to be more complex and less disciplined. Establishing the lengths first makes the following steps more tractable.

6 POSITIONS OF THE ZEROES AND PARTITIONING THE WORD FORMS

In the previous step, the procedure made an educated guess about the desired lengths of the stems and affix parts. If the guess produces poor solutions, then the procedure backtracks and modifies the guess. For the time being, however, the procedure sticks to the assumption made in the previous section and adds some zeroes as necessary so that the lengths of the word forms meet the lengths required by the partition in the Table 3. The procedure adds the required amount of zeroes in all different permutations. Thus, from now on, the procedure has full tables of our example words (with zeroes). Each table conforms with the lengths but has the zeroes in random positions of the word forms. There may be many such tables and it is not practical to enumerate them before filtering out the clearly impossible ones.

A human linguist would perhaps immediately see the positions where the zeroes are best added, e.g., **koir0issa**, because in this way the letters in the first four positions of the stem would be identical in all stems of the lexeme. The linguist would exclude other positions for the zero because they would lead to an unnatural correspondence of letters. However, a computer procedure can manage with many possible versions where a correct number of zeroes are added but perhaps not in the correct places.

The partitioning of the word forms into stems and affixes is done by adding a fixed number of boundary symbols (+) into the word forms. If we have only suffixes (as in our example) or only prefixes, one boundary will be sufficient. If we have both prefixes and suffixes, two boundary symbols are needed. More than two may be needed for Semitic languages with both prefixes and suffixes and even interdigitation (where vowel affixes are inserted inside the word root). In fact, the interdigitation does not cause any additional problems for the two-level or rewrite rules cf., e.g., Kataja and Koskenniemi (1988). The problem with interdigitation is how to build the lexical or underly-

Table 4:
Example word forms
with zeroes and boundaries
in bad positions

SgNom	SgPtv	SgIne	PlPtv	PlIne
talo +	talo + a	talo + ssa	talo + ja	talo + issa
särö +	särö + ä	särö + ssä	särö + jä	särö + issä
risti +	risti + ä	risti + ssä	riste + jä	riste + issä
lov**i** +	lov**e** + a	lov**e** + ssa	lov**i** + a0	lo0**v** + issa
kala +	kala + a	kala + ssa	kalo + ja	kalo + issa
k**o**ira +	k**o**ira + a	k**o**ira + ssa	0**k**oir + ia	0**k**oir + issa

ing representation out of morpheme-like elements. That is not, however, within the scope of rule discovery. The added boundary symbols split the word form into even- and odd-numbered segments. The stem of the lexeme consists of either the odd-numbered or even-numbered segments. The remaining segments represent the inflectional affixes which belong to the grammatical form.

On the basis of the calculation of the lengths in the previous steps (and knowing at which end the affixes are located), we can insert the boundary symbols at uniquely determined positions in the word forms which have been augmented with zeros as necessary. This is now done for all alternative tables. If our assumption on the positions of the affixes is wrong, then the following steps will produce poor results, and we must backtrack and revise our assumption.

In our example, the procedure adds exactly one boundary symbol to each of these four word forms and one of the alternative tables for our example could now look as shown in Table 4. There, some corresponding letters are quite incompatible with each other, e.g., **i-e-e-i-v** in the fourth position of the stems for **lovi**, and **o-k** in the second position of the stems for **koira**. We consider vowels to be incompatible with consonants (except with semivowels) and *vice-versa*. We exclude all tables which violate this coarse constraint. Therefore Table 4 (and other tables containing equally poor correspondences) will be excluded.

Among the possibilities, the procedure also produces Table 5, where the characters in the corresponding positions of the same stems are reasonably congruent with each other. The same holds for characters in the corresponding positions of the suffixes. A human linguist would probably like this table best.

SgNom	SgPtv	SgIne	PlPtv	PlIne
talo +	talo + a	talo + ssa	talo + ja	talo + issa
särö +	särö + ä	särö + ssä	särö + jä	särö + issä
risti +	risti + ä	risti + ssä	riste + jä	riste + issä
lovi +	love + a	love + ssa	lov0 + ia	lov0 + issa
kala +	kala + a	kala + ssa	kalo + ja	kalo + issa
koira +	koira + a	koira + ssa	koir0 + ia	koir0 + issa

Table 5:
Example word forms with added zeroes and boundaries in good positions

The simple example we are studying would have only a few thousand different possibilities for adding the two required zeroes. The coarse checking of impossible tables would be no problem at all. With larger examples, some planning for the efficient exclusion of the impossible tables is needed. One option is to use a kind of branch-and-bound algorithm to prune the search space more efficiently. Adding a zero to a wrong place often causes an impossible correspondence so that one can exclude a whole class of tables before even creating them. Another alternative would be to represent the set of tables as finite-state networks which would remain quite reasonable in size and be straightforward to construct. Impossible paths (representing tables) would be excluded by a sequence of XFST or FOMA rules which would filter out strings (i.e. tables) by checking the compatibility of each character position. The PhD dissertation of Grzegorz Kondrak gives, among other things, a survey of the various methods which have been used in aligning words – see Kondrak (2002).

7 MORPHOPHONEMES AND THE REPRESENTATION OF MORPHEMES

Now that we have processed the initial matrix of word forms into a reasonably small set of tentative tables differing from each other in the positioning of zeroes, the next step of the procedure is to rank the remaining tables according to the morphophonemes that they imply. The different stems (for each lexeme) and affixes (for each grammatical form) in the Table 5 are now of equal length. Stems can be extracted and aligned as in the Table 6, where the bottom row indicates the morphophonemic representation that follows from the aligned

Table 6:
Stems of the lexemes i.e. word forms with boundaries and zeroes but the affix part removed

t a l o	s ä r ö	r i s t i	l o v i	k a l a	k o i r a
t a l o	s ä r ö	r i s t i	l o v e	k a l a	k o i r a
t a l o	s ä r ö	r i s t i	l o v e	k a l a	k o i r a
t a l o	s ä r ö	r i s t e	l o v 0	k a l o	k o i r 0
t a l o	s ä r ö	r i s t e	l o v 0	k a l o	k o i r 0
t a l o	**s ä r ö**	**r i s t ie**	**l o v ie0**	**k a l ao**	**k o i r a0**

Table 7:
Affixes of the grammatical forms

SgPtv	SgIne	PlPtv	PlIne
a	s s a	j a	i s s a
ä	s s ä	j ä	i s s ä
ä	s s ä	j ä	i s s ä
a	s s a	i a	i s s a
a	s s a	j a	i s s a
a	s s a	i a	i s s a
aä	**s s aä**	**ij aä**	**i s s aä**

stems. Most positions in the series of stems contain the same character. The vowels at the end alternate a bit in some stems. The relations between corresponding letters in the endings also look regular (see the Table 7 where we, again, have included in the last row the morphophonemic representations of the affixes).

The criterion for ranking the alternative tables is based on the quality of morphophonemes that each table implies. We denote the morphophonemes by indicating the characters they represent, e.g., **ie**, **ie0**, **ao**, **a0**, **aä** and **ij**.[5] According to common linguistic knowledge, similar phonemes are more likely to alternate with each other and radically different ones may not alternate with each other.

In Table 8, we see the (coarse) phonemic characterisations for the Finnish vowels according to the features used in the IPA (International Phonetic Alphabet). In addition to the named features, we have associated (somewhat *ad hoc*) numerical values with the features for the purposes of the discovery procedure. We can approximate vowels in the languages of the world according to the IPA using three

[5] Technically, e.g., **ie** stands for the tuple (i, i, i, e, e) and **ie0** for $(i, e, e, 0, 0)$, cf. footnote in Section 3.

Letter	IPA	Height		Backness		Rounding	
ä	/æ/	(near-)open	1	front	1	unrounded	0
a	/ɑ/	open	1	back	5	unrounded	0
e	/e/	close-mid	5	front	1	unrounded	0
ö	/ø/	close-mid	5	front	1	rounded	1
o	/o/	close-mid	5	back	5	rounded	1
i	/i/	close	7	front	1	unrounded	0
y	/y/	close	7	front	1	rounded	1
u	/u/	close	7	back	5	rounded	1
j	/j/	semivowel	9	front	1	unrounded	0

Table 8: Phonological features and numerical approximations of Finnish vowels

	Front		Back	
Height	unrounded	rounded	unrounded	rounded
Close	/i/ i	/y/ y		/u/ u
Close-mid	/e/ e	/ø/ ö		/o/ o
Open	/æ/ ä		/ɑ/ a	

Table 9: Distinctions in the Finnish vowel system

digits (as in Table 8): one for the tongue height with a scale from 1 (low or open) to 7 (high or close), a second for the backness with a scale from 1 (front) to 5 (back), and third for rounding with 0 (unrounded) and 1 (rounded). The values represent just an ordinal scale, not any physical dimensions. A tongue height of 5, for instance, is higher than 1 but not necessarily five times as high.[6] Phonemes in most languages employ only a part of the possible heights and backness values.

There is no opposition between open and near-open vowels in Finnish, so the difference between them is ignored and the value 1 used for both. The Finnish vowel system is often represented as in Table 9. According to the Tables 8 and 9, **ie** makes a perfect morphophoneme for our purposes, as these two vowels differ by one feature only, the height of the tongue (and even only by one step).

[6] Furthermore, the front vowels differ in their backness: /**i**/ and /**y**/ are most front, /**e**/ and /**ø**/ a bit less front and /**æ**/ even more to the back. These differences play no role in the present discussion. See *http://www.langsci.ucl.ac.uk/ipa/* for more information on the IPA alphabet.

Table 10: Phonological features of common Finnish consonants

Letter	IPA	Place		Manner	Voicing	
m	/m/	bilabial	1	nasal	voiced	1
p	/p/	bilabial	1	plosive	unvoiced	0
v	/v/	labiodental	2	fricative	voiced	1
t	/t/	alveolar	4	plosive	unvoiced	0
d	/d/	alveolar	4	plosive	voiced	1
s	/s/	alveolar	4	fricative	unvoiced	0
r	/r/	alveolar	4	trill	voiced	1
l	/l/	alveolar	4	lateral approximant	voiced	1
j	/j/	palatal	7	approximant	voiced (semivowel)	1
ng	/ŋ/	velar	8	nasal	voiced	1
k	/k/	velar	8	plosive	unvoiced	0
h	/h/	pharyngeal	10	fricative	unvoiced	0

Using this table, the components of **aä** differ only by one feature: **ä** being front and the **a** being back. In **ao** there are two minimally different values: **o** is rounded and one step more close than **a** which is unrounded. In Finnish, there is no back vowel more like **a** than **o**. Note that the semivowel **j** is given a characterisation both as a vowel (in Table 8) and as a consonant (in Table 10).

Consonants have more possible feature values than vowels. The place of articulation corresponds to the backness of vowels, but the different manners of articulation are less related to each other and do not form a continuum or an ordinal scale. Voicing is a binary feature and can be represented in the same way as the rounding of vowels. The features of some Finnish consonants are given in Table 10. The similarity between **i** and **j** requires a bit of linguistic knowledge: palatal consonants are pronounced roughly at the same place as front vowels, and that a semivowel is like a vowel but pronounced with some friction. The numerical values for the backness of vowels and the place of articulation for consonants seem to be on different scales. No vowels are articulated as front as some consonants. When comparing **j** with vowels, we may treat it as: tongue height 9 (more closed than any vowel), backness 1 (i.e. front), unrounded 0, as in Table 8.

Morpho-phoneme	Heights	Backnesses	Roundings	Penalty
aä	1,1	5,1	0,0	1
ij	7,9	1,1	0,0	1
ie	7,5	1,1	0,0	1
ie0	7,5,–	1,1,–	0,0,–	3
ao	1,5	5,5	0,1	2
a0	1,–	5,–	0,–	2

Table 11:
Penalties for differences of phonemes in morphophonemes implied by Table 5

SgNom	SgPtv	SgIne	PlPtv	PlIne
talo +	talo + a	talo + ssa	talo + ja	talo + issa
särö +	särö + ä	särö + ssä	särö + jä	särö + issä
risti +	risti + ä	risti + ssä	riste + jä	riste + issä
lovi +	love + a	love + ssa	lovi + 0a	lovi + 0ssa
kala +	kala + a	kala + ssa	kalo + ja	kalo + issa
koira +	koira + a	koira + ssa	koiri + 0a	koiri + 0ssa

Table 12:
Example word forms with added zeroes and boundaries in alternative (almost good) positions

Tables organised according to articulatory features reflect the closeness of phonemes. Phonological or morphophonological alternations typically modify just one or sometimes two features of a sound such as the voicing of a stop or the backness of a vowel. For the purposes of the discovery procedures, no perfect metric is required. A rough approximation will be sufficient if it is capable of excluding linguistically infeasible alternations.

In Table 11, we list the numerical characteristics of the vowels in the morphophonemes and use an ad hoc formula for computing a penalty. For the vowel height, we use four levels: (near-)open, close-mid, close, and semivowel. One-level difference in height, a different backness, or rounding counts as 1 each; a bigger difference in height, or if a zero belongs to the morphophoneme that corresponds to a deletion, then it counts as 2. The total penalty of the morphophonemes in Table 11 is 10.

Some other aligned tables may have survived when we filtered out the impossible alignments. In fact, a fairly good alternative would be the one given in Table 12. This one differs from our earlier good table by inserting the zeroes one position later than in our earlier good alter-

Table 13: Penalties for differences of phonemes in morphophonemes implied by Table 12

Morpho-phoneme	Heights	Backnesses	Roundings	Penalty
aä	1,1	5,1	0,0	1
j0	7,–	1,–	0,–	2
i0	7,–	1,–	0,–	2
ie	7,5	1,1	0,0	1
ao	1,5	5,5	0,1	2
ai	1,7	5,1	0,0	3

native. The alignment is almost as good as the earlier one. If we add up the penalties, as in Table 13, we get a total penalty of 11. With these weightings, the procedure would choose the "right" solution, but in general, we cannot select the best one on the basis of the morphophonemes alone. The "almost good" solution could be used for deducing the rules and would account for the facts. Linguistically, however, **ai** is not a very attractive alternation because the components are from the extreme ends of both the height and backness scales. Maybe the penalty for such ought to be even higher than 3.

8 HOW A LINGUIST COULD FIND THE RULES FOR MORPHOPHONEMES

The procedure has used the suitability of induced morphophonemes in order to guide the selection of paradigm tables for the next step where two-level rules are deduced. The procedure continues with the best alternative (which was presented in Table 5). The morphophonemes established in the preceding steps already define how they may be realised on the surface level. The rules must specify in what kinds of contexts each of the alternatives can occur. We first discuss how a human linguist could approach the discovery of the two-level rules needed.

We drop the pluses from our table as they are not needed in the example we are studying (and not in many other cases either). The alphabet of the two-level grammar is already defined through the letters occurring in the example and as a consequence of the morphophonemes implied by the alignment:

```
Alphabet a k l o r s t v ä ö  aä:a aä:ä ij:i ij:j
  ie:i ie:e ie0:i ie0:e ie0:0 ao:a ao:o a0:a a0:0 ;
```

Table 14 presents the facts about the contexts where the morphophonemes occur. The pair in focus (one in each word form) is marked with bold face. Different realisations of a morphophoneme are given in separate columns. For the convenience of the reader, the columns have been arranged so that the realisation with incoherent surrounding contexts is always in the leftmost column, whereas the realisations which have more regular contexts are listed in the other columns.

The upper half of the middle column shows that certain stem-final vowel morphophoneme realisations (**ie:e**, **ie0:0**, **ao:o** and **a0:0**) may only occur before a plural affix which starts with **i:** or **ij:**. Looking at the other columns, we see that such contexts do not occur with the other realisations of the morphophonemes (i.e. **ie:i**, **ie0:0**, **ao:a**, **a0:a** or **ie0:i**). Thus those realisations of stem-final vowels occur (**ie:e**, **ie0:0**, **ao:o** and **a0:0**) only in this context, and this context is the only alternative. As two-level rules:

```
"ie"   ie:e <=> _ [i: | ij:] ;
"ie0" ie0:0 <=> _ [i: | ij:] ;
"ao"   ao:o <=> _ [i: | ij:] ;
"a0"   a0:0 <=> _ [i: | ij:] ;
```

In the rightmost column, we have one more realisation for which we (as linguists) find a simple formulation: **ie0** is realised as **i** at the end of a word form. As a two-level rule:

```
"ie0" ie0:0 <=> _ .#.: ;
```

When we look (as linguists) at the distribution of the plural **ij:j** alternative, we note that the stem has to end in a surface vowel. If the end vowel disappears on the surface, then **ij:j** may not occur, thus:

```
"ij" ij:j <=> [:o | :ö | :e ] _ ;
```

The last morphophoneme to account for is **aä**. Whereas the earlier cases depended only on the immediate context, here the realisation depends on all vowels of stem to the left. The backness of the vowels is decisive, and we note that in the middle column there is at least

Table 14: Realisations of morphophonemes and their contexts

r i s t **ie:i**		
r i s t **ie:i** aä:ä	r i s t **ie:e** ij:j aä:ä	
r i s t **ie:i** s s aä:ä	r i s t **ie:e** i s s aä:ä	
l o v **ie0:e** aä:a	l o v **ie0:0** ij:i aä:a	l o v **ie0:i**
l o v **ie0:e** s s aä:a	l o v **ie0:0** i s s aä:a	
k a l **ao:a**		
k a l **ao:a** aä:a	k a l **ao:o** ij:j aä:a	
k a l **ao:a** s s aä:a	k a l **ao:o** i s s aä:a	
k o i r **a0:a**		
k o i r **a0:a** aä:a	k o i r **a0:0** ij:i aä:a	
k o i r **a0:a** s s aä:a	k o i r **a0:0** i s s aä:a	
l o v ie0:0 **ij:i** aä:a	t a l o **ij:j** aä:a	
k o i r a0:0 **ij:i** aä:a	s ä r ö **ij:j** aä:ä	
	r i s t ie:e **ij:j** aä:ä	
	k a l ao:o **ij:j** aä:a	
s ä r ö **aä:ä**	t a l o **aä:a**	
s ä r ö s s **aä:ä**	t a l o s s **aä:a**	
s ä r ö ij:j **aä:ä**	t a l o ij:j **aä:a**	
s ä r ö i s s **aä:ä**	t a l o i s s **aä:a**	
r i s t ie:i **aä:ä**	l o v ie0:e **aä:a**	
r i s t ie:i s s **aä:ä**	l o v ie0:e s s **aä:a**	
r i s t ie:e ij:j **aä:ä**	l o v ie0:0 ij:i **aä:a**	
r i s t ie:e i s s **aä:ä**	l o v ie0:0 i s s **aä:a**	
	k a l ao:a **aä:a**	
	k a l ao:a s s **aä:a**	
	k a l ao:o ij:j **aä:a**	
	k a l ao:o i s s **aä:a**	
	k o i r a0:a **aä:a**	
	k o i r a0:a s s **aä:a**	
	k o i r a0:0 ij:i **aä:a**	
	k o i r a0:0 i s s **aä:a**	

one back vowel somewhere in the stem whereas in the left column no back vowels occur, thus:[7]

```
"aä" aä:a <=> [:a | :o | :u] ?:?* _ ;
```

It would be difficult even for the linguist to generalise the contexts in the leftmost column. Therefore, we say that the realisations in that column are the *default realisations* of those morphophonemes. As they are the only remaining alternatives of their morphophonemes, no rules are needed for them.

The linguist who knows some facts about Finnish notices that the inflectional class of **risti** is productive and contains plenty of nouns, whereas that of **lovi** is a closed class containing fewer words. The steps above resulted in different stem-final morphophonemes for the lexical representations these words. The establishment of the lexical representations and the design of the corresponding rules was not affected by the existence of two apparently-overlapping inflectional classes. When one builds a lexicon, one must decide, for each such an ambiguous noun, to which class it belongs in order to build an appropriate lexicon entry. That decision may be made by human informants (e.g., by crowdsourcing), or by using evidence, e.g., from corpora or Internet search engines.

9 PROCEDURE FOR FINDING SHORT CONTEXT

We have now seen how a human linguist might discover the two-level rules according to the envisaged procedure. The formal procedure handles each morphophoneme separately. The procedure could start, e.g., with **ie** and try to find a phonologically natural characterisation of contexts such that **ie:i** in our data occurs in contexts of that type but **ie:e** does not. If a satisfactory result is not reached, the procedure tries to find a natural set of contexts where the other alternative **ie:e** may occur but **ie:i** may not.

Many morphophonological alternations are conditioned by an immediate context consisting of just one or a few phonemes. Thus, the procedure tries to find as short a context as possible which still dis-

[7] In a slightly larger and thus more realistic example we would also have forms like **s i n ie0:e aä:ä** and **k a s t ie:i aä:a** which would rule out attempts to explain the outcome of **aä** on the basis of the stem-final vowel alone.

criminates the desired realisations from all other realisations of the morphophoneme (cf., e.g., Theron and Cloete, 1997). The procedure starts with the full contexts for **ie:i** where both the left and right contexts are present, and we have added a word boundary symbol **#:0** at the beginning and end of the word forms:

```
#:0 r i s t _ #:0
#:0 r i s t _ aä:a #:0
#:0 r i s t _ s s aä:ä #:0
```

This disjunction of the full contexts clearly separates the occurrences of **ie:i** from the other realisation (**ie:e**). The procedure drops characters from the outer ends as long as the disjunction still separates the occurrences. If possible, the longer side of the context is truncated before the shorter side. When processing the above contexts, the procedure will erase the left context altogether but one character must be left to the right context – resulting in a context _ **[#: | aä:ä | s]**. The procedure does not accept this result, as it contains both vowels and consonants and, therefore, is not acceptable on the same arguments which were mentioned when the some morphophonemes were excluded as unnatural in Section 7.

The procedure tests the other alternative **ie:e**, and shortens the contexts until there is just one character left in the right context, i.e. _ **[ij:j | i]**. This context is acceptable on the same grounds as the morphophoneme **ij** itself. Some cleaning and generalisation may still be needed as explained in the next section.

For morphophoneme realisations **ie0:e**, **ao:a**, and **a0:a**, the procedure fails to find a natural context which would discriminate them from the other alternatives. For the other realisations of these morphophonemes, the procedure succeeds in the same way as for **ie:e**. For **ie0:0**, the smallest discriminating context clearly becomes _ **#:0**.

The contexts for the realisation **ij:i** would be just one character to the left, **[ie0:0 | a0:0]** _, and for **ij:j** similarly **[o | ö | ie:e | ao:o]** _. If more than one alternative explanation remains, later steps will decide which one is preferred over the others.

10 GENERALISING CONTEXTS

In the previous section, the procedure found two-level contexts consisting of character pairs. In most cases (but not always), having both levels is superfluous. Proper discrimination can usually be achieved using either the lexical or surface context. Furthermore, the contexts can (and ought to) be generalised to use whole classes of phonemes instead of listing only those letters and combinations which happened to be present in the examples. Both the lexical and surface contexts are generalised first, and the choice between them is made thereafter.

The naturalness of contexts can be evaluated according to the phonemes which occur in individual positions of contexts. Morphophonemes occurring in lexical contexts are treated by splitting them into their component letters, e.g., **ao** is treated as if **a** and **o** would occur in that position of the contexts. Table 15 shows how the phonological properties of one-character-long left contexts of the morphophoneme **ij** can be summarised.

Table 15: Generalising the one character left context for **ij:j**

contexts for ij:j				
two-level	ao:o ie:e o ö	Height	Back	Round
lexical	a e i o ö	1–7	1–5	0–1
surface	e o ö	5	1–5	0–1
contexts for ij:i				
two-level	ie0:0 a0:0	Height	Back	Round
lexical	0 a e i	1–7	1–5	0
surface	0	–	–	–
ignoring 0		Place	Manner	Voicing
surface	r v	2–4	trill/fricative	1

Using the table, the procedure observes that the lexical contexts for the two morphophonemes are overlapping and therefore the lexical context is not useful, but the surface contexts are able to discriminate between the two alternants. If we study the surface contexts, we see that surface zeroes probably ought to be ignored, i.e. the context letter would be the one preceding (or, respectively, following) a zero. In this particular case, it would provide a reasonable context for **ij:i**, but it does not matter because we get a better one from the alternative.

The surface contexts in our examples for **ij:j** allow many vowels, and the procedure generalises it to allow all vowels because the context discriminates between the alternatives. This turns out to be fortunate as further examples would bring stems ending in **u** or **y**.

Let us look back at Table 12, our second best table with zeroes and boundaries. That one was only a bit worse when measured on the basis of the morphophonemes it implied. The context condition for the plural **ij** for the best table (after generalisations) was any surface vowel. The plural morpheme in the second best table needs two separate morphophonemes instead of one, **j0** and **i0**, and a rule for each where the context conditions would be no simpler than in the best table. The stem-final morphophonemes **ai** and **ei** for the second best table would be problematic, as both surface realisations of the morphophonemes occur in an identical surface context. Remember that we decided to omit the explicit boundary symbol (+) for brevity. If we keep the boundary symbol, the rules of the best table can use even the surface context, but definitely not the rules of the second best table. With a reasonable penalty formula for rules and their context expressions, the best table would again get a better score than the second best.

The example of Finnish nouns was a very restricted one because it contained no alternating consonants. Finnish is known to be a language with a fairly complex morphophonology, and has plenty of consonantal alterations as well. Consonant gradation weakens voiceless stops **k**, **p** and **t**. A distance metric for consonants would be similar to that of vowels. The place of articulation can be expressed on a scale from 1 to 11 and the voicing with values 0 and 1 as in Table 10. The manner of articulation often varies in the alternations. However, the different manners are in no particular order and it is difficult to say which manners are close to (or far away from) each other.

Let us consider a mini example of the Finnish consonant gradation in Table 16.

Table 16: Examples of consonant gradation of **p** with added zeroes and boundaries

	SgNom	SgEss	SgIne	SgAll
‘twig’	varpu +	varpu + na	varvu + ssa	varvu + lle
‘shield’	kilpi +	kilpe + nä	kilve + ssä	kilve + lle
‘pond’	lampi +	lampe + na	lamme + ssa	lamme + lle
‘stick’	keppi +	keppi + nä	kep0i + ssä	kep0i + lle

The procedure can readily accept the induced morphophonemes **pv**, **pm** and **p0** because they are articulated approximately in the same place: **pv** in places 1–2, **pm** in places 1–1. The procedure aligns and forms the morphophonemes for this mini example without problems. Note that the direction of the change is well motivated: the stop **p** becomes a bit more like the immediately preceding phoneme **r**, **l** or **m**.

The task for the procedure is to find a generalised context which would account for the occurrence of the alternative realisations. Again, left hand context is not useful at all. The right-hand context appears to be quite sufficient. The lexical context for the the weak alternatives **p:v**, **p:m**, and **p:0** is **u s s aä**, **e s s aä**, **i s s aä** or **u l l e**, **e l l e**, **i l l e** which can be generalised as **V: C: C:** (i.e. a closed syllable) without losing any discriminative power.

11 PROCEDURE FOR FINDING HARMONY CONTEXTS

For the morphophoneme **aä**, the mechanism of finding a short context fails, and the procedure *knows* that it has failed (cf. Table 14). The shortening does not progress successfully, and all tentative contexts are unacceptable (having consonants and vowels in the same positions).

Harmony or agreement can be detected using phonological features. The data for this purpose is collected in Table 17, where the vowel context or the whole preceding word form is summarised. On the left, the word forms with different vowel configurations are listed. In the middle, each of these vowels is represented numerically according to Table 8. On the right, there is a summary of the set of vowels in the word form indicating the range of tongue height, backness and rounding for that word form.

The procedure looks at these summary ranges, and tests each of them whether some of these ranges could be used to separate the words where **aä:ä** occurs from those where **aä:a** occurs. Height is not able to discriminate, and neither can rounding. Backness clearly can. A criterion requiring that a word has at least one vowel with backness > 1 will indicate all those contexts or words where **aä:a** may occur.

The procedure thus finds a positive criterion for the occurrences of **aä:a**, but for **aä:ä** there is only a negative criterion. Thus, the pro-

Table 17: Harmony in terms of tongue height, backness and rounding of vowels in the word form

aä:ä		Height	Back	Round
särö_, säröss_, säröj_	(2,1,0), (4,1,1)	2–4	1	0–1
säröiss_	(2,1,0), (4,1,1), (7,1,0)	2–7	1	0–1
risti_	(7,1,0), (7,1,0)	7	1	0
ristej_	(7,1,0), (5,1,0)	5–7	1	0
risteiss_	(7,1,0), (5,1,0), (7,1,0)	5–7	1	0

aä:a		Height	Back	Round
talo_, taloss_, taloj_	(1,5,0), (5,5,1)	1–5	5	0–1
taloiss_	(1,5,0), (5,5,1), (7,1,0)	1–7	1–5	0–1
love_, lovess_	(5,5,1), (5,1,0)	5	1–5	0–1
lovi_, loviss_	(5,5,1), (7,1,0)	5–7	1–5	0–1
kala_	(1,5,0), (1,5,0)	1	5	0
kaloj_	(1,5,0), (5,5,1)	1–5	5	0–1
kaloiss_	(1,5,0), (5,5,1), (7,1,0)	1–7	1–5	0–1
koira_, koirass_	(5,5,1), (7,1,0)	5–7	1–5	0–1
koiri_, koiriss_	(5,5,1), (7,1,0), (7,1,0)	5–7	1–5	0–1

cedure classifies **aä:ä** as the default realisation and no rule is written for it. For **aä:a** the criterion can be expressed as:

```
"aä" aä:a <=> [:a | :o | :u] ?:?* _ ;
```

where the procedure generalises the backness even to **u** – for which there is no example in the data.

12 FINDING GRAMMATICAL CONDITIONS

The above methods do not handle morphophonemic alternations which are difficult or impossible to describe using phonological contexts alone. Grammatical conditioning of morphophonemic alternations is a common phenomenon, though. In some languages, certain inflectional forms are characterised by alternations in the stem rather than by overt affixes. The alignment, adding zeroes and segmenting possible affixes, would proceed with no special problems, but no rules would be found.

The discovery procedure sketched above could easily be modified to discover grammatically conditioned regularities. Suppose that we create a special symbol for each grammatical form. We would include an appropriate special symbol at the end of each word form. If all other patterns for contexts fail, then those special symbols would be tried as contexts. If the presence of certain symbols would discriminate the occurrences of different realisations, the procedure would output a rule of the following type:

```
"ae:e"  ae:e <=> _ ?:?* Passive: ;
```

where **Passive** stands for such a special symbol (which is required to be somewhere in the right context).

13 COMPILING AND VERIFYING THE RULES

Even before we have an implementation for the discovery procedure, one can simulate the steps of the procedure manually. The resulting two-level rules may be compiled using the open source two-level rule compiler HFST-TWOLC written by Miikka Silfverberg of the HFST team at the University of Helsinki (Silfverberg and Lindén, 2009).

The table of example word forms after the addition of zeroes and morpheme boundaries can be used for testing the (automatically or manually produced) two-level rules by a trivial conversion script which reads in the aligned word forms, builds the morphophonemes, and outputs strings of character pairs suitable for the HFST-PAIR-TEST program of the HFST suite. The file would have lines such as the following:

```
t a l o
t a l o ij:j aä:a
k a l ao:o ij:j aä:a
```

These lines can be input to the HFST-PAIR-TEST program which reports any violations against the rules it finds in the test data. Violations pinpoint the example word form, the position and the rule where the mistake appears to be.

In addition to testing the obvious positive examples, one may produce a file which systematically contains negative examples, i.e. examples derived from the correct ones but where at least one rule is violated. Such a set of negative examples can be produced out of the positive ones by (1) creating a transducer E which accepts the positive examples, (2) taking its input (i.e. upper) side $E.u$, (3) computing the transducer P which accepts the pair alphabet of the rules, (4) computing $[E.u\ .o.\ P*] - E$ and (5) listing the pair strings it accepts.

14 CONCLUSIONS AND FUTURE TASKS

A concrete implementation of the discovery procedure would, of course, be needed in order to draw any final conclusions. The paper is intended to be a useful specification for implementing the process.[8]

One goal of the above discussion has been to explain the utility of the phoneme-by-phoneme alignment of word forms. If an acceptable alignment is reached, then the establishment of lexical representations is trivial and the induction of rules is fairly simple. Different types of (morpho)phonological phenomena may need specialised functions which can be added in a modular fashion. The two-level grammars that the procedure creates can cope with some phenomena which often cause problems for linguists when they are writing rules:

- *Interactions* do not occur between rules, except that one has to process the realisations of each morphophoneme together so that the morphophoneme leaves just one of its surface realisations without a rule (as the default realisation). The inference of each rule is entirely independent of the form (or existence) of other rules.
- *Epenthesis* (where surface phonemes have no counterpart in the lexical representation) never occurs. Instead, the alignment produces a morphophoneme, e.g., **e0** if a vowel **e** is inserted to resolve a complex consonant cluster. Technically, epenthesis is reduced to normal correspondences. (Rules for epenthesis are tricky in some formalisms.)
- *Overlapping* or inclusive *contexts*. Finnish consonant gradation has one such example with the weak counterparts of **k**. A single **k** is normally deleted in the weak grade (**koko** – **koon**), but in some

[8] There are suitable open source tools available which support the algebra of weighted finite-state transducers and which could be used for implementing the discovery procedure, see, e.g., Lindén *et al.* (2011).

words it alternates with **v** (**puku** – **puvun**) and between identical vowels preceded by a long vowel it is spelled as an apostrophe (**raaka** – **raa'an**). The alignment produces three different morphophonemes (**k0**, **kv** and **k'**) for these cases and the rule discovery notices no problems at all.

- All conditions for a certain surface realisation of a morphophoneme are represented using a single rule. In this way, the rest of the sources for rule interactions are avoided. It should be noted that some natural phonological contexts may consist of disjoint parts. The above discussion did not cover such cases. Combinations of contexts can be expressed using multiple context parts in the two-level rules (provided that each context part can be discovered separately), but the discovery of rules needing multiple contexts is not discussed in this paper.

Some simplicity and computational feasibility was gained by adopting the above framework (with many morphophonemes) at a price of losing some linguistic elegance. The procedure moves a part of the complexity of the morphophonology from the rules into the lexical representations. The rules need not bother with all possible realisations of an underlying phoneme because, instead of a phoneme, there is a morphophoneme which specifies exactly what the alternatives are. It should be noted that the above procedure (and the two-level rules in general) appear to work best with *phonemic alphabets*. Conventional orthographies may be quite different from their pronunciation and thus complicate rule discovery. Furthermore, isolating languages like Chinese or Vietnamese have little morphophonology to be discovered by any procedure. In some European languages, the morphophonological rules play a minor role. Thus, the proposed procedure might be most useful in languages with plenty of regular phonological alternations.

The tractability of the procedure is not obvious, even if we assume that the table of examples is well chosen. Some care has been taken in order to restrict the searching sufficiently so that the complexity of the computation would not explode. It seems to be useful to fix the assumptions concerning the stem and affix lengths before one starts finding appropriate places for the zeroes. Similarly, it seems useful to check for rough compatibility before generalising any contexts. These and other similar precautions do not affect the end result, but may help in finding the solutions reasonably fast.

A linguist would probably like to merge similar or related morphophonemes. Good candidates for merging would be morphophonemes which have a default realisation in common. Merging usually requires some revision of the corresponding two-level rules. It appears to be fairly straightforward to check whether such mergers can be done while keeping the revised rules simple and deducible. The criteria used for deducing single rules apply as such for the merging of rules. Such generalisations would make the lexical representations of morphemes and affixes simpler while remaining fully equivalent with the initial version. These tasks and questions may possibly be studied and solved by future work, as well as by the elaboration and tuning of the penalty scores for rules.

REFERENCES

Kenneth R. BEESLEY and Lauri KARTTUNEN (2003), *Finite State Morphology*, Studies in Computational Linguistics, 3, University of Chicago Press, additional info, see: www.stanford.edu/~laurik/fsmbook/home.html.

Erwin CHAN (2008), Structures and Distributions in Morphology Learning, a dissertation in Computer and Information Science, University of Pennsylvania.

Mathias CREUTZ and Krista LAGUS (2004), Induction of a Simple Morphology for Highly-Inflecting Languages, in *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology*, pp. 43–51, Association for Computational Linguistics, Stroudsburg, PA, USA.

Daniel GILDEA and Daniel JURAFSKY (1995), Automatic induction of finite state transducers for simple phonological rules, in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pp. 9–15, Association for Computational Linguistics, Cambridge, Massachusetts.

John GOLDSMITH (2006), An Algorithm for the Unsupervised Learning of Morphology, *Natural Language Engineering*, 12(4):353–371.

Sharon GOLDWATER and Mark JOHNSON (2004), Priors in Bayesian Learning of Phonological Rules, in *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology*, pp. 35–42, Association for Computational Linguistics, Stroudsburg, PA, USA.

Mans HULDEN (2009), Foma: a Finite-State Compiler and Library, in *Proceedings of the Demonstrations Session at EACL 2009*, pp. 29–32, Association for Computational Linguistics, Stroudsburg, PA, USA, http://www.aclweb.org/anthology/E09-2008.

Mans HULDEN, Iñaki ALEGRIA, Izaskun ETXEBERRIA, and Montse MARITXALAR (2011), Learning word-level dialectal variation as phonological replacement rules using a limited parallel corpus, in *Proceedings of EMNLP 2011, Conference on Empirical Methods in Natural Language Processing*, DIALECTS'11, Association for Computational Linguistics, Stroudsburg, PA, USA.

Mark JOHNSON (1984), A Discovery Procedure for Certain Phonological Rules, in *Proceedings of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, pp. 344–347, Association for Computational Linguistics, Stroudsburg, PA, USA, `http://www.aclweb.org/anthology/P84-1070`.

Lauri KARTTUNEN (1993), Finite-state Constraints, in *Proceedings of the International Conference on Current Issues in Computational Linguistics, June 10–14, 1991. Universiti Sains Malaysia, Penang, Malaysia*, pp. 173–194.

Lauri KARTTUNEN and Kenneth R. BEESLEY (2001), A short history of two-level morphology, `http://www.helsinki.fi/esslli/evening/20years/twol-history.pdf`.

Lauri KARTTUNEN, Kimmo KOSKENNIEMI, and Ronald M. KAPLAN (1987), A compiler for two-level phonological rules, in M. DALRYMPLE, R. KAPLAN, L. KARTTUNEN, K. KOSKENNIEMI, S. SHAIO, and M. WESCOAT, editors, *Tools for Morphological Analysis*, volume 87-108 of *CSLI Reports*, pp. 1–61, Center for the Study of Language and Information, Stanford University, Palo Alto, California, USA.

Laura KATAJA and Kimmo KOSKENNIEMI (1988), Finite-state Description of Semitic Morphology: A Case Study of Ancient Accadian, in *COLING Budapest: Proceedings of the 12th Conference on Computational Linguistics*, pp. 313–315, Association for Computational Linguistics, Stroudsburg, PA, USA, `http://aclweb.org/anthology-new/C/C88/C88-1064.pdf`.

Grzegorz KONDRAK (2002), *Algorithms for Language Reconstruction*, Ph.D. thesis, University of Toronto.

Kimmo KOSKENNIEMI (1983), *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*, number 11 in Publications, University of Helsinki, Department of General Linguistics.

Kimmo KOSKENNIEMI (1984), A General Computational Model for Word-Form Recognition and Production, in *Proceedings of COLING-84, 2–4 July 1984, Stanford University, California*, pp. 178–181, Association for Computational Linguistics, Stroudsburg, PA, USA.

Kimmo KOSKENNIEMI (1991), A Discovery Procedure for Two-level Phonology, in L. CIGNONI and C. PETERS, editors, *Computational Lexicology and Lexicography: Special Issue Dedicated to Bernard Quemada*, volume VI:I, Giardini editori e stampatori in Pisa, Pisa, Italy.

Krister LINDÉN, Erik AXELSON, Sam HARDWICK, Tommi A. PIRINEN, and Miikka SILFVERBERG (2011), HFST – Framework for Compiling and Applying Morphologies, in C. MAHLOW and M. PIOTROWSKI, editors, *Systems and Frameworks for Computational Morphology 2011 (SFCM-2011)*, volume 100 of *Communications in Computer and Information Science*, pp. 67–85, Springer-Verlag.

Tom M. MITCHELL (1982), Generalization as search, *Artificial Intelligence*, 18(2):203–226.

Kemal OFLAZER, Sergei NIRENBURG, and Marjorie MCSHANE (2001), Bootstrapping morphological analyzers by combining human elicitation and machine learning, *Computational Linguistics*, 27(1):59–85.

Kemal OFLAZER and Sergei NIRENBURG (1999), Practical Bootstrapping of Morphological Analyzers, in *Proceedings of Computational Natural Language Learning (CoNLL99). Workshop at EACL'99*, pp. 143–146, Springer-Verlag.

José ONCINA, Pedro GARCÍA, and Enrique VIDAL (1993), Learning subsequential transducers for pattern recognition interpretation tasks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):448–458.

Miikka SILFVERBERG and Krister LINDÉN (2009), Conflict Resolution Using Weighted Rules in HFST-TWOLC, in *Proceedings of the 17th Nordic Conference of Computational Linguistics, NODALIDA 2009*, pp. 174–181, Northern European Association for Language Technology (NEALT), `http://hdl.handle.net/10062/9752`.

Pieter THERON and Ian CLOETE (1997), Automatic Acquisition of Two-Level Morphological Rules, in *Fifth Conference on Applied Natural Language Processing Proceedings of the Conference*, pp. 103–110, Association for Computational Linguistics.

David TOURETZKY, Gillette ELVGREN, and Deirdre W. WHEELER (1990), Phonological rule induction: An architectural solution, in *Proceedings of the 12th Annual Conference of the Cognitive Science Society (COGSCI-90)*, pp. 348–355, Cognitive Science Society.

David YAROWSKY and Richard WICENTOWSKI (2000), Minimally supervised morphological analysis by multimodal alignment, in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics: Hong Kong*, Association for Computational Linguistics.

This work is licensed under the Creative Commons Attribution 3.0 Unported License.
`http://creativecommons.org/licenses/by/3.0/`