Technical Report

UCAM-CL-TR-794
ISSN 1476-2986

Number 794

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Grammatical error prediction

Øistein E. Andersen

January 2011

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted 2010 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Girton College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Summary

In this thesis, we investigate methods for automatic detection, and to some extent correction, of grammatical errors. The evaluation is based on manual error annotation in the Cambridge Learner Corpus (CLC), and automatic or semi-automatic annotation of error corpora is one possible application, but the methods are also applicable in other settings, for instance to give learners feedback on their writing or in a proofreading tool used to prepare texts for publication.

Apart from the CLC, we use the British National Corpus (BNC) to get a better model of correct usage, WordNet for semantic relations, other machine-readable dictionaries for orthography/morphology, and the Robust Accurate Statistical Parsing (RASP) system to parse both the CLC and the BNC and thereby identify syntactic relations within the sentence. An ancillary outcome of this is a syntactically annotated version of the BNC, which we have made publicly available.

We present a tool called GenERRate, which can be used to introduce errors into a corpus of correct text, and evaluate to what extent the resulting synthetic error corpus can complement or replace a real error corpus.

Different methods for detection and correction are investigated, including: sentence-level binary classification based on machine learning over *n*-grams of words, *n*-grams of part-of-speech tags and grammatical relations; automatic identification of features which are highly indicative of individual errors; and development of classifiers aimed more specifically at given error types, for instance concord errors based on syntactic structure and collocation errors based on co-occurrence statistics from the BNC, using clustering to deal with data sparseness. We show that such techniques can detect, and sometimes even correct, at least certain error types as well as or better than human annotators.

We finally present an annotation experiment in which a human annotator corrects and supplements the automatic annotation, which confirms the high detection/correction accuracy of our system and furthermore shows that such a hybrid set-up gives higher-quality annotation with considerably less time and effort expended compared to fully manual annotation.

Preface

First of all, there are many people who deserve to be mentioned on this page, but whose names do not appear — from my parents, who hardly ever lost their patience when I bombarded them with questions as a child, to the visiting friend who presented me with the idea of a doctorate over dinner at a time when further studies did not seem an obvious choice —, and I should like to express my gratitude to every single one and assure them that their various contributions have not been forgotten.

The person most directly involved in the work presented in the following is of course my supervisor, Ted Briscoe, who has always been available, under whom I have enjoyed considerable freedom, and who has at the same time led me back on track whenever questions of only tangential relevance to my research topic started to take up too much of my time.

Parts of the work has been done in collaboration with Andreas Vlachos and Jennifer Foster, and the annotation experiment could not have been successfully carried out without Diane Nicholls' kind co-operation. Rebecca Watson and John Carroll have both been very helpful with technical issues related to the CLC and RASP.

Many useful comments have been provided by conference attendees and anonymous reviewers, but also by colleagues in the research group, including my second supervisor Ann Copestake, and by friends and acquaintances from other fields.

This work would not have been possible without the support from Cambridge ESOL[1] and the Cambridge Overseas Trust.

Finally, I should like to thank Helen Yannakoudakis for having read through the thesis in its entirety and contributed to making it more readable than it would otherwise have been, as well as my examiners, Stephen Pulman and Paula Buttery, for their thorough and friendly approach and valuable comments and suggestions, many of which have resulted in improvements to this document.

[1]) This dissertation reports on research supported by the University of Cambridge ESOL Examinations.

Table of contents

Chapter 1.

Introduction

The study of grammar can be traced back to ancient India, where it originated as an auxiliary discipline to the study of the Vedas, the oldest sacred texts of Hinduism. Amongst the first Sanskrit grammarians were Sakatayana (*c.* 8th c. BC) and Yaska (*c.* 7th c. BC), and the discipline culminated with the completion of Panini's (*c.* 5th c. BC) grammar known as अष्टाध्यायी (8 chapters; *Aṣṭādhyāyī*), a description of Sanskrit morphology in 3,959 rules. Prominent 20th-century linguists influenced by Panini's ideas include Saussure and Chomsky, not to mention Bloomfield, who went so far as to characterise Panini's grammar as 'one of the greatest monuments of human intelligence' (1933, p. 11), though its reputation as a concise yet exhaustive description of Sanskrit, whose perfection remains unsurpassed by any other grammar for any language, may be exaggerated (Mishra 1986). Unfortunately, early Indian insights into grammar remained unknown in Europe until the 19th century.

In the Hellenistic world, the need for linguistic study arose with the development of rhetoric and logic. Concepts like 'syllables' and 'sentences' were already being discussed in the 5th century BC by the Sophists, one of whom, Protagoras of Abdera, is said to have pointed out the distinction between the three genders of Greek: masculine, feminine and neuter (Aristotle *Rhet.*, III.v, p. 1407b). According to Plato (428/7–348/7 BC), his teacher Socrates (*c.* 469–399 BC) observed the distinction between two different kinds of words, ῥήματα (words) and ὀνόματα (names), from which λόγοι (speech) are made up (*Crat.*, p. 431b), defined more precisely in another of Plato's dialogues as denoting actions and the ones who perform them, respectively (*Soph.*, p. 262a). Plato's pupil Aristotle (384–322 BC) notably recognised a verb not in the present tense or a noun/adjective not in the nominative case as a πτῶσις (fall) of the primitive ῥῆμα or ὄνομα (*Int.*, ii–iii, p. 16b), and also used the term σύνδεσμοι (bonds) to refer to words which do not denote actions or actors but whose function it is to bind the sentence together (Aristotle *Rhet.*, III.v, p. 1407a).

Grammar started to emerge as a separate discipline with the Stoics (4th c. BC onwards), who distinguished between a word *per se,* the mental image it evokes and the 'thing' or 'situation' to which it refers. At this point, ἄρθρα (joints) (articles and pronouns) were separated

from the remaining σύνδεσμοι (conjunctions and prepositions), and the Stoics provided more accurate and appropriate definitions of the different μέρη (parts of speech) λόγου and introduced new distinctions, including the one between active, middle and passive voices. The study of language continued with the Alexandrians (3rd c. BC onwards), whose grammatical insights were epitomised in the Τέχνη (*Ars*) Γραμματική (*Grammatica*) (*c.* 100 BC), attributed to DIONYSIUS of Thrace (*Ars Gr.*). This first extant grammar of Greek is mainly a condensed treatise on morphological categories, defining eight parts of speech (*viz,* ὄνομα (noun), ῥῆμα (verb), μετοχή (participle), ἄρθρον (article), ἀντωνυμία (pronoun), πρόθεσις (preposition), ἐπίρρημα (adverb) and σύνδεσμος (conjunction)) and their accidents (*e.g.,* case, number, person, mood and tense). Whether Dionysius' treatment of the subject 'have been little improved upon for more than twenty centuries' (DINNEEN 1967, p. 150) or whether 'the continuity of the classical tradition [...] often [lie] in the names of the categories rather than in their contents' (MICHAEL 1970, p. 490) is a matter for another dissertation, as is the question of correct attribution, exact chronology and relative interdependence of the first Græco-Latin grammars, but it seems safe to say that the ideas expressed in Dionysius' τέχνη has had a lasting influence on the formulation of grammars for European languages until this day: smaller Latin *artes grammaticæ* are very similar to it in both form and contents; Apollonius Dyscolus (*fl.* AD 100) assumed the same categories for his extensive treatises on Greek grammar, including syntax, whereupon Priscian's (*fl.* AD 500) *ars* was based; and subsequent grammarians largely followed the precedent thus established, also for the study of vernacular languages rather remote from Greek and Latin morphology and syntax.

One problem with adopting the categories initially discovered in the Greek language, instead of reapplying the methodology used to determine the relevant ones in the first place, is that the description of one language could easily be influenced by specificities of another: for instance, Priscian retained the optative mood alongside the subjunctive, despite there being no morphologically distinct optative verb forms in Latin; and BULLOKAR's first English grammar (1586), largely modelled on what is generally known as *Lily's Latin grammar,* kept the case system almost entirely unchanged, only unifying dative and ablative into 'gainative', thus characterising English as a language with five cases. Despite shortcomings such as these, categories first established over two millennia ago are still commonly used, in particular for the description of European languages.

1.1. Rules of grammar

More relevant to our concerns is the appearance of the concept of grammatical and correct language. Arguably, any description of a language has the potential to become normative, in the same way as Panini's grammar was to become the very definition of classical Sanskrit, forms and constructions not explicitly mentioned therein thus effectively being outlawed. The word ἑλληνίζειν (speak Greek correctly) is attested in PLATO (*Prot.*, p. 328[a]); his predecessor Socrates regarded 'correct language [as] the prerequisite for correct living (including an efficient government)'

(Guthrie 1971, p. 276); and the preoccupation with proper use of language emerged even earlier amongst the Greeks. Τέχνη γραμματικὴ originally signified the modest art of combining γράμματα (letters) into words (*cf.* Plato *Crat.*, p. 431[e]); by Dionysius' time, it had extended to encompass the 'knowledge of the language generally employed by poets and writers' (*Ars Gr.*, I) in its entirety (including, *e.g.*, etymology and literary criticism); but narrower definitions ultimately prevailed:

> Grammatica est scientia recte loq̃ēdi.
>
> — Isodore *Orig.*, I.v, p. 4[r]

> Grammatica eſt recte ſcribendi atque loquendi ars.
>
> — Lily *c.* 1500

> Gramatica. Arte, che'nſegna a correttamente parlare, e ſcriuere.
>
> — Accademia 1612

> Grammaire ſ.f. L'art qui enſeigne à parler & à eſcrire correctement.
>
> — Académie 1694

> Grammar is the Art of rightly expreſſing our thoughts by Words.
>
> — Lowth 1762, p. 1

> Grammar has sometimes been described as the Art of speaking and writing correctly. But people may possess the Art of correctly using their own language without having any knowledge of grammar. We define it therefore as the Science which treats of words and their correct use.
>
> — West 1894, p. 35

> Grammar [. . . :] a person's knowledge and use of a language.
>
> — Hornby 2005

In practice, 'grammar' was long synonymous with Latin grammar as laid down by Priscian *et alii,* who dealt mainly with morphology and syntax, and is therefore sometimes used to denote these subdisciplines only, as is indeed the case for many of the grammar books mentioned in this chapter. Apart from such instances, however, the terms 'grammar' and 'grammaticality' will in the following assume a more general sense related to linguistic knowledge and ability as suggested by the quotations above; in other terms, a 'grammatical' error may consist not only in a morphologically malformed word or a syntactically incorrect construction, but also for instance in a non-idiomatic expression or a confusion between similar words. (Certain low-level errors like accidental spelling mistakes should perhaps not be regarded as grammatical errors *per se,* but it is difficult to maintain a sharp delineation, and their absence would in any case be a prerequisite for grammaticality.)

1.2. Codification of English grammar

The state of the English language started to become a preoccupation in the 16th century, after the Reformation, as the vernacular was gradually replacing Latin as the language of learning. BULLOKAR's first English grammar presented itself as being

> ſufficient for the ſpedi lærning how too pärc Engliſh ſpech for the perfecter wrýting thær-of, and uzing of the beſt phráſéz thær-in,
>
> — 1586

and CAWDREY's *Table Alphabeticall,* the first monolingual English dictionary (limited to 'hard words', but including English definitions), claimed to

> conteyn[...] and teach[...] the true vvriting, and underſtanding of hard vſuall Engliſh wordes, borrowed from the Hebrew, Greeke, Latine, or French, &c.
>
> — 1604

A century and a half later, many more dictionaries and grammar books had been published, but no norm seemed to gain general acceptance; a writer of English had

> neither *Grammar* nor *Dictionary*, neither Chart nor Compaſs, to guide [him] through this wide ſea of Words.
>
> — WARBURTON 1747, p. xxv, original emphasis.

Italy and France had meanwhile established academies, each of which had produced an authoritative dictionary and provided definitive advice *re* correct use of language, whereas the need for codification of the English tongue was instead to be fulfilled by two seminal works published just after the middle of the the 18th century: Johnson's dictionary in 1755, and Lowth's grammar in 1762. Samuel Johnson compiled, in the space of nine years, what 'easily ranks as one of the greatest single achievements of scholarship' (BATE 1978), containing over 40,000 entries and 110,000 literary examples, a novelty at the time. He 'left, in the examples, to every authour his own practice unmoleſted' (JOHNSON 1755, p. A4), and he claimed that he did

> not form, but regiſter the language; [...] not teach men how they ſhould think, but relate how they ha[d] [t]hitherto expreſſed their thoughts.
>
> — *ibid.*, p. C2

Nevertheless, his approach was clearly not purely descriptive, as illustrated by his 'adjuſting the *Orthography*, which ha[d] been to th[at] time unſettled and fortuitous' (*ibid.*, p. A3, original emphasis), evoking etymology and analogy as guiding principles in addition to established usage. Also Robert LOWTH quoted the best writers of his time, but often to point out that they had 'fallen into miſtakes, and been guilty of palpable errors in point of Grammar' (1762, p. ix), for he considered that

> pointing out what is wrong [...] m[ight] perhaps be found [...] to be [...] the more uſeful and effectual manner of inſtruction.
>
> — *ibid.*, p. x–xi

The discipline has seen progress since: a large number of dictionaries and grammar books have been developed for a variety of purposes; Johnson's dictionary has been replaced by the *Oxford English Dictionary* as the reference; descriptive grammars from around 1900 (including, *e.g.,* Pedersen's) have provided a more complete description of English; a large number of different syntactic theories have been developed; modern grammars like HUDDLESTON & PULLUM's have attempted 'to bridge the large gap [...] between traditional grammar and the partial descriptions [...] proposed by [linguists]' (2002, p. xv); and the increasing availability of corpora has made it possible to investigate actual usage quite precisely, thus to some extent obviating the need to rely on intuition and unilateral judgement. More revealing, perhaps, is the continuity: English orthography has changed very little in over two hundred and fifty years; rules first formulated by Lowth have now become universal (*e.g.,* the ban on double negatives and the use of *were* rather than *was* with *you*), though others have since fallen out of favour; the literati rather than οἱ πολλοὶ influence the language sanctioned by dictionaries and grammar books; and the continued existence of grammar/style guides like Fowler's *Modern English Usage* attests on a sustained quest for the Holy Grail of 'propriety and accuracy' (LOWTH 1762, p. ix).

CAWDREY's dictionary was aimed 'for the benefit & helpe of Ladies, Gentlewomen, or any other vnskilfull perſons' (1604) who had not benefited from a classical education, and BULLOKAR intended his grammar not only to be useful for the Englishman learning his own and other languages, but also to be 'very-aid-fųl tœ the ſtrańġǫr ţœ lærń ėngliſh perfectļy and spėdiļy' (1586). The first monolingual English dictionary aimed exclusively at foreigners did however not appear until 1942, when Hornby's predecessor of the Oxford *Advanced Learner's Dictionary* was published in Japan; since then, the number of dictionaries and grammars of this type has multiplied, and Cambridge University Press now asserts that its pedagogical *English Grammar in Use* is 'the world's best-selling grammar book'.

1.3. Grammaticality and acceptability

LOWTH claimed to be the first to use negative examples, in which case he may also have pioneered the binary division between grammatical and ungrammatical sentences:

> The principal deſign of a Grammar [...] is to teach us to expreſs ourſelves with propriety [...], and to be able to judge of every phraſe and form of conſtruction, whether it be right or not. The plain way of doing this, is to lay down rules, and to illuſtrate them by [positive and negative] examples.
>
> — 1762, p. x

CHOMSKY presented an apparently identical goal almost two centuries later:

> The fundamental aim in the linguistic analysis of a language *L* is to separate the *grammatical* sequences which are the sentences of *L* from the *ungrammatical* sequences which are not sentences of *L*.
>
> — 1957, p. 13, original emphasis

As pointed out by SAMPSON, however, traditional grammarians were merely concerned with the 'uninteresting' set of ungrammatical sentences that people are actually seen to produce, including those which will be labelled as socially deprecated rather than ungrammatical by agnostic or non-prescriptive linguists, as opposed to Chomsky's perhaps more ambitious aim of dividing the universe of combinatorically possible sentences into two (2007). A different perspective is provided by BURT & KIPARSKY in the context of learner errors:

> It is common practice for transformational grammarians to prefix any ungrammatical sequence of words with an asterisk (*). These can include sentences that no one would say. Since we are only interested in sentences actually spoken or written by people learning [English as a second language], we will draw the distinction by prefixing spoken ungrammatical sentences [...] with the dagger (†).
>
> — 1972, p. 2

A similar separation is often made between sentences that are *grammatical* in the sense that they are licensed by a formal grammar, and the ones that are *acceptable* or 'actually grammatical, *i.e.*, acceptable to a native speaker' (CHOMSKY 1957, p. 13), which is useful for the task of evaluating the extent to which a given grammatical formalism corresponds to a native speaker's idea of language. This distinction between grammaticality and acceptability becomes harder to define when the grammar under consideration is the one assumed to exist within the speaker's mind. An oft-quoted example of a putatively grammatical albeit clearly unacceptable construction is a sentence with deep recursive centre-embedding:

(1) *The certain reputation the reactions the sentences the grammar the syntactician overwhelmed by clever ideas dreamt up in his dusty study located in the attic generated as grammatical and his students disparaged as abhorrently complex caused earned him was to become his only solace in old age.*

Whether the rarity, not to say inexistence, of such highly convoluted constructions in practice should be ascribed to rules in the mental grammar or to linguistically more peripheral factors, such as memory limitations, is of little importance to our concerns. The same applies to sentences which seem more intuitively syntactically well-formed, but hardly acceptable (at least not outside a somewhat contrived context):

(2) ?*Colo[u]rless green ideas sleep furiously.*[2]

[2]) *ibid.*, p. 15.

Such sentences are sometimes said to lack 'semantic soundness', though the conditions determining whether a given sentence is semantically sound or not, especially in the absence of context, are at best difficult to lay down.

Apart from the special case of evaluating the adequacy of a grammatical formalism, the need for a distinction between grammaticality and acceptability seems to be felt most acutely for constructions which do not occur naturally and will therefore not be employed in the following.

Ungrammatical sentences obviously include syntactically erroneous sentences:

(3) **Little a boy the ran street up.*[3]

Incorrect constructions of the type often brought about by translation also result in ungrammaticality:

(4) **I am here since two years.*

Overly colloquial constructions will typically be considered ungrammatical as well (keeping in mind that this depends on the context, and that overly formal constructions can of course be equally inappropriate):

(5) **Him and her don't want no cake.*

Then, there are less clear cases like poetic constructions, the acceptability of which depends on the context:

(6) *?And all the air a solemn stillness holds.*

This leads us to the issue of grammaticality as a gradient:

> The degree of grammaticalness is a measure of the remoteness of an utterance from the [...] set of perfectly well-formed sentences.
>
> — Chomsky 1961, p. 237

Linguists typically use stigmata made up of asterisks and question marks to indicate different levels of grammaticality; there is no standard system, but the ordering from grammatical to ungrammatical is usually consistent with Andrews' classification:

[3]) Ex. 3–6 taken from Quirk & Svartvik 1966, p. 10.

√	Fullkomlega tæk og eðlileg	(Completely acceptable and natural)
?	Tæk, en kannski svolítið óeðlileg	(Acceptable, but perhaps somewhat unnatural)
??	Vafasöm, en kannski tæk	(Doubtful, but perhaps acceptable)
?*	Verri, en ekki alveg ótæk	(Worse, but not totally unacceptable)
*	Algjörlega ótæk	(Thouroughly unacceptable)
**	Hryllileg	(Horrible)

— 1990, p. 203, original translations

In this thesis, the asterisk (*) indicates a clear instance of ungrammaticality, and the question mark (?), a dubious instance.

Sorace & Keller showed correlation between experimental grammaticality judgements and the violation of grammatical rules (2005): violation of a 'hard constraint' (*e.g.,* agreement or inversion) causes a higher degree of ungrammaticality than violation of a 'soft constraint' (*e.g.,* definiteness or verb class constraints); furthermore, each additional violation decreases the degree of grammaticality, but the violation of several soft constraints typically results in less severe ungrammaticality than the violation of one hard constraint. This dichotomous point of view is controversial, but the idea that certain rules are more important than others seems intuitively plausible and has appeared in many variations. The potential consequences of gradient phenomena in grammar for syntactic theories and grammatical formalisms are however clearly outside the scope of this thesis (see, *e.g.,* Aarts 2007 and Manning 2003 for more on those aspects). More relevant to our concept of grammaticality is Lennon's observation with respect to the following sentence produced by a German university student of English during a semester in England (shown with a possible correction):

(7) a. ?*There is a dam wall which should protect the village from flood.*

b. *There is a dam which is meant to protect the village against floods/flooding.*

In his opinion, 'it may be various infelicities occurring in close proximity which persuade the native speaker he or she has recognized an error' (1991). On the one hand, this implies that minor errors in the sense of slightly infelicitous expressions should not be regarded as erroneous as long as they are few and far between, which seems reasonable given that neither native nor non-native speakers always choose the most conventional expression or systematically recognise this kind of mild deviance; on the other hand, and more importantly, not even slight infelicities can be disregarded completely, given that they may add up to a noticeable error.

We shall return to the issue of grammatical errors, including an operational definition and different ways in which to classify them, in Chapter 2.

1.4. Corporal evidence

The idea of regarding works of approved authors as a model worthy of emulation is millennia old and has led grammar writers and dictionary compilers to collect innumerable examples to serve as 'reservoir[s] of linguistic usage' (KUČERA 1967), an activity which reached its pre-digital pinnacle with the millions of quotation slips gathered in a purpose-built scriptorium and put into pigeonholes by James Murray during the preparation of the *Oxford English Dictionary*. Digital corpora can be seen as a natural continuation of this tradition, furthermore allowing linguistic enquiry to be founded more directly on real language data, as opposed to the intuition and experience of a well-read man supported by occasional quotations.

> The basis [for writing a grammar of English] must be copious materials, made up of continuous stretches or 'texts' taken from the full range of co-existing varieties and strata of educated English, spoken as well as written, at the present time.
>
> — QUIRK 1960

The value of corpora was not immediately appreciated by all schools of linguists, though, as seen in Robert Lee's dismissal of Nelson Francis and Henry Kučera's emerging efforts to create what was to become the Brown University *Standard Corpus of Present-Day Edited American English, for use with Digital Computers:*

> That is a complete waste of your time and the government's money. You are a native speaker of English; in ten minutes you can produce more illustrations of any point in English grammar than you will find in many millions of words of random text.
>
> — FRANCIS 1979

CHOMSKY went so far as to reject naturalistic data as actively misleading:

> Any natural corpus will be [wildly] skewed. Some sentences won't occur because they are obvious, others because they are false, still others because they are impolite.
>
> — 1962

The alternative to corpus data is introspection, relying on a native speaker's ability not only to establish whether or not a given sentence is grammatical, but also to pull out relevant examples from a bottomless reservoir within his own mind, as it were. Though he be able to do this, the problem of representativity nevertheless remains; the need for external input may be more obvious to the lexicographer than to the syntactician, but both run the risk of parochialism:

> [Native speakers'] linguistic activity ranges from writing love letters or scientific lectures to speaking upon a public rostrum or in the relaxed atmosphere of a private dinner party. Since native speakers include lawyers, journalists, gynæcologists, school teachers, engineers, and a host of

> other specialists, it follows [...] that no grammarian can describe adequately the grammatical and stylistic properties of the whole repertoire from his own unsupplemented resources: 'introspection' as the sole guiding star is clearly ruled out.
>
> — Svartvik & Quirk 1980, p. 9

This is not the place for a lengthy discussion of the respective merits and demerits of corporal and noetic sources of linguistic evidence. It seems clear to us, though, that corpora are crucial to the enterprise of grammatical error detection in at least two ways: a reference corpus allows us to verify that our idea of grammaticality corresponds to the language produced by those who are considered knowledgeable (*e.g.*, reputable authors), and an error corpus makes it possible to check that commonly committed errors are handled correctly. Such collections are not 'millions of words of random text'; rather, a linguistic corpus is

> a collection of texts assumed to be representative of a given language, or other subset of a language, to be used for linguistic analysis.
>
> — Francis 1979

Issues related to collection, annotation and use of error corpora will be discussed further in Chapter 2. Questions of corpus encoding and exploitation in general will be treated in Chapter 3, which includes information on the parsed version of the Cambridge Learner Corpus and describes the process of parsing the British National Corpus

Finally, whereas corpora are generally stored electronically for convenience and ease of processing, at least one sizeable printed corpus in the sense of a carefully selected collection of printed matter has existed: Kaeding's variety of lexical and graphemic frequency counts was derived manually by a large team of people from a corpus of 11 million words of text of all genres, ranging from original and translated poetry to military orders and personal correspondence (1897). The amount of manual labour required obviously made such projects few and far between.

1.5. Automatic error detection

With the advent of the computer, rudimentary proofreading became an automatable task, and many people today might be frightened by the idea of having to prepare an important document without any sort of electronic support or 'validation' of their writing. As we shall see in Chapter 2, however, popular spelling and grammar checkers, useful though they may be to someone who is aware of their limitations and knows the language well, are still no substitute for either subeditors or language teachers.

In Chapter 4, we look at the task of detecting ungrammatical sentences as a binary classification problem using a set of correct and incorrect sentences as training data, and the

performance is compared to a system that aims at detecting individual instances of specific types of error. The development of further specialised classifiers will be described in Chapters 6 and 7.

A prerequisite for many of the error detection techniques is a sample of correct as well as incorrect language in an easily exploitable format. Any corpus is finite and thus deficient in the sense that it does not explicitly exemplify all possible constructions. The idea of creating synthetic errors will be explored in Chapter 5.

Other partial solutions to this problem can be to supplement an error corpus like the Cambridge Learner Corpus with a larger corpus of correct language like the British National Corpus, to use encyclopædic or taxonomic resources like WordNet, and to cluster words in order to explicate latent information from the corpus. Ways to circumvent the data sparsity problem will be the main topic of Chapter 6.

The problem of insufficiently comprehensive corpora can of course be alleviated, albeit not fully solved, by developing larger corpora. For an error-annotated corpus in particular, the amount of manual work required during its preparation is considerable, and partial automation could potentially make the process significantly more efficient. In Chapter 7, we shall see how a set of error detection techniques combined with a dedicated annotation tool can make the annotation process more efficient and contribute towards more consistent annotation than completely manual procedures. This constitutes both an application of error detection and correction techniques and a means by which to obtain larger and better error corpora which in turn can make those techniques more effective, and thus provides a natural conclusion to the work presented in this thesis.

Chapter 2.

Right and wrong

In this chapter, the concept of grammatical error will be discussed in more detail, in terms of what constitutes an error as well as how to classify different types. We shall discuss two tasks for which a definition of error is paramount: error annotation in corpora and automatic error detection as a writer's tool. The two applications are related in other ways as well: a good writer's tool should take into account the types of error typically committed, information which is latent in an error-annotated corpus; conversely, the annotation task can be carried out with the help of an automatic tool, an aspect to which we shall return in Chapter 7.

Corder identified three stages in error analysis: recognition, linguistic description and psycho-linguistic explanation (1974). Both automatic error detection tools and error annotation schemes tend to provide a description in terms of a category or a suggested correction, whereas neither usually includes a higher-level explanation, partly because the necessary information is unavailable.

Exactly what constitutes an error is a controversial topic, but at the same time crucial for annotation, identification being a prerequisite. Our concept of error is close to James's provisional definition of 'a language error as an unsuccesful bit of language' (1998, p. 1). This includes in particular the subclass of errors *(sensu lato)* variously termed mistakes, slips or *lapsi calami,* namely deviances which the writer would have been able to correct himself if someone had drawn his attention to them; this is convenient for error annotation since the writer's intention cannot normally be established for instances of deviant language in a corpus, and one of the strengths of automatic error detection tools is exactly that they can draw the writer's intention to differences between what he wrote and what he meant to write.

Lennon defined an error committed by a non-native speaker as

> a linguistic form or combination of forms which, in the same context and under similar conditions of production, would, in all likelihood, not be produced by the speakers' native speaker counterparts.
>
> — 1991

This provides a useful complement to James's vague definition by pointing out that it would be inappropriate to compare spontaneous speech to prepared oration or an informal e-mail to an acclaimed piece of literature. This consideration is of course not limited to non-native speakers, but rather a consequence of the fact that the language produced by a native speaker varies with the individual and the situation and does not conform to a single standard. A similar warning against applying inappropriate criteria of correctness was expressed by NICHOLLS (pertaining to a written learner corpus):

> [When in doubt,] ask yourself whether a native English speaker of average educational level might conceivably write the same thing.
>
> — 2007

CORDER pointed out that a sentence can be superficially deviant, contextually/situationally inappropriate or both; moreover, even a well-formed and appropriate utterance may be erroneous by not conveying the intended meaning (1974). Superficial deviance, which amongst idealised native speakers only occurs as the result of inattention, can generally be identified without additional information of any kind; contextual or situational inappropriateness, on the other hand, ensues from lack of knowledge not of the language itself but of how it is used in a particular context or situation, and this type of error can only be detected when sufficient information about the context or situation is available, which may not always be the case for corpus data or text checked by an automatic error detection tool; finally, utterances that are 'right by chance' (*ibid.*) can only be identified as erroneous if the speaker's/writer's intention can be established to differ from the actual meaning of the utterance produced, which is usually not feasible in the scenarios considered here. Thus, in a perfectly annotated written error corpus, all instances of superficial deviance and some instances of contextual inappropriateness will have been identified, whereas no superficially correct and contextually appropriate utterances will have been found to be erroneous, although, in all likelihood, a certain proportion of them are; the same limitation of available evidence applies to an automatic error detection tool.

2.1. Error corpora

Already Lowth used authentic examples to illustrate common errors, a practice elevated to something of an art form by the brothers FOWLER, who explicitly chose 'to pass by all rules [...] that are shown by observation to be seldom or never broken' (1906, p. iii), relying on contemporary literature and journalism to select 'blunders' worthy of mention. In a similar vein, language teachers at all levels more or less conciously/explicitly take errors committed by pupils past and present into account when they plan the next lesson, the knowledge thus accumulated by experienced teachers in turn being crucial for the development of teaching materials for a given demographic.

To our best knowledge, the study of errors committed by professional writers — and, more generally, native speakers from all walks of life in different writing situations — remains sporadic and unsystematic; no serious effort seems to have been made to characterise with any level of precision the types of error typically found in either edited or unedited text (apart from some studies on keyboard-related typographic errors, *e.g.*, MacNeilage 1964, as well as smaller-scale collections akin to the one included in Foster 2004), leaving authors of writer's guides and developers of software tools with no option but to rely on intuitive and impressionistic approaches. Significant practical and technical hurdles would have to be overcome to obtain a representative and sufficiently large sample of unedited text in particular, not to mention the appointment of a team of experts with sufficient authority to add the appropriate amount of proverbial red ink. On the other hand,

> the conviction that a native speaker of a language knows enough about it not only to teach it to others but to become an expert at designing functional computer aids [...has led to] a considerable oversupply of inferior products,
>
> — Kučera 1992

focusing not on areas proven to be difficult for the intended users, but rather on non-errors such as 'the great American bugaboo, [...] the schoolmarm dogma that passives are bad' (*ibid.*), a superstition that could easily be overcome by studying a corpus of correct and successful language.

It has been argued that non-native speakers should be regarded as competent users of their own, personal idiosyncratic dialect, whereby constructions that sound ungrammatical to native ears should not be labelled as such, since 'they are in fact *grammatical* in terms of the learner's language' (Corder 1971, original italics). Needless to say, moving the goalposts to a different field altogether is utterly unhelpful for someone who wants to improve his foreign-language skills[4]; it may well be true that a thorough analysis of 'the learner's language' *qua* language can be illuminating when subsequently compared to the grammar of the 'real' language, but a reverence for individual differences does not seem wholly justified in this case given that a learner's grammar will often be internally inconsistent in ways geographical dialects are not, and more importantly that a learner's unintentional deviation from the norm is quite different from a dialect speaker's adherence to a different norm. A more valid criticism is that a lop-sided focus on errors obscures what the non-native speaker gets right as well as more subtle linguistic deviation not normally covered under the concept of error, or

[4]) The idea that '[g]rammatical accuracy is not always essential for accurate communication' (Page 1990) has affected language pedagogy, and those who merely want to acquire the bare necessities of linguistic competence in order to make themselves understood might be well advised not to focus too much on errors that are not crucial for the message to be conveyed effectively. Such considerations are however orthogonal to our concerns; an error detection tool is aimed at helping authors who actively want to avoid errors, and an annotated error corpus is useful for someone who would like to investigate the communicative effects of different types of error.

> that work on the linguistic description of learners' languages can be seriously hindered or sidetracked by [...] the 'comparative fallacy'.
>
> — Bley-Vroman 1983

Partly to avoid this problem, but also since many errors would not be apparent in isolated words or sentences, non-native error corpora are not limited to erroneous words or sentences, but include entire texts with correct as well as incorrect attributes. The term *learner corpus* may therefore be more appropriate, despite the high error rate generally observed.

2.2. Learner corpora

Corpora have been used for language teaching purposes since their inception; vocabulary frequency information extracted from the Brown corpus was pointed out as potentially useful already by Francis (1967). Studying the properties of edited text does not provide much insight into the difficulties faced by those who are still learning the language, though, and perhaps even less so in the case of foreign learners:

> For language teaching [...,] it is not only essential to know what native speakers typically say, but also what the typical difficulties of the learners of a certain language, or rather of certain groups of learners of this language, are.
>
> — Nesselhauf 2004

The obvious solution — if, like Allén, we value 'authenticity' and find it 'reasonable to take a look at real manifestations of language when discussing linguistic problems' (1992) — is to use a corpus of language written by learners as the basis of such knowledge.

Akin to 'normal' corpora, learner corpora have to be constructed according to strict design criteria in order to be representative, in this case of a precisely delimited population of learners. A learner corpus usually contains texts not only in different genres but written by learners from different backgrounds under different conditions, which means that additional metadata is required; Granger *& al.* (2007) and Ellis & Barkhuizen (2005, p. 30) mention variables such as age, sex, mother tongue, social background, task type, topic, time limitations and use of reference tools. It is thus possible to extract texts written by a given group of learners or investigate differences between groups, but this fragmentation also means that the full corpus must be much larger than one aimed at being representative of one type of text produced by one homogeneous group, such as the level of written business English attained by female French monolingual learners of English at the end of a Scientific Baccalaureate.

A large number of English learner corpora have appeared during the last few years; Schiftner provided an overview of over two dozen corpora ranging in size from a few thousand words to tens of millions of words, incorporating texts written by learners from one or

multiple linguistic backgrounds and including different amounts of annotation, mostly developed at academic institutions (2008, *q.v.* for essential characteristics and references; see also PRAVEC 2002 for further details on the larger/older corpora). The majority of the corpora contain fewer than one million words, and only three are larger than five million words: the 30-million-word Hong Kong University of Science and Techonology (HKUST) Corpus and two commercial corpora collected to be used for dictionary compilation and textbook writing by Longman (10 mill. words) and Cambridge University Press (30 mill. words). The Cambridge Learner Corpus (CLC)[5] includes material written during language examination by learners of English at different levels and from all over the world; half of it has been manually error-coded and will be used as a realistic source of errors throughout this thesis.

2.3. Error classification

Perhaps the oldest error categories still in use are the ancient Greek concepts of *barbarism,* a word or form corrupted by foreign influence, and *solecism,* originally a faulty concord. These terms do not seem to be much in favour by English linguists nowadays, but such a coarse-grained classification is sometimes useful, as seen in FOSTER's definition of grammatical error (*sc.* solecism) as an erroneous construction made up from individually correct words (2004), as opposed to misspellings (*sc.* barbarisms) detected by naïve spelling checkers. For error annotation, more detailed categories are needed if a distinction between different types of error is to be made at all.

2.3.1. Surface structure taxonomies

Dictionaries of errors are organised alphabetically, at least on a superficial level, although extensive cross-referencing may allow a more systematic approach than one might at first suspect. Other similar books are explicitly divided into separate sections corresponding to different types of error, such as FITIKIDES' classification into five major categories (1936):

1. Misused forms (wrong preposition, wrong tense, etc.);
2. Incorrect omission (missing preposition, missing auxiliary, missing morpheme such as plural *-s* or past tense *-ed*, etc.);
3. Unnecessary words (superfluous article, superfluous *to*, etc.);
4. Misplaced words (*e.g.,* adverbials);
5. Confused words (wrong preposition [*sic*], wrong noun, etc.).

[5]) http://www.cambridge.org/elt/corpus/learner_corpus2.htm

Similar surface structure taxonomies have been discussed more recently, for instance by DULAY, BURT & KRASHEN, who noted that '[l]earners may *omit* necessary items or *add* unnecessary ones; they may *misform* items or *misorder* them' (1982, p. 150, original italics), or by JAMES, who preferred a quintipartite division into omissions, overinclusions, misselections, misorderings and blends (1998, p. 111). Surface structure taxonomies have been critisised for not taking into account the modern view of language as fundamentally hierarchical rather than as a concatenation of words like beads on a string, but surface structure is still used for error annotation, partly because it provides a practical approach, but it has also been argued that the concept of surface structure is psycholinguistically sound in the sense that language users often think of errors in such terms (*e.g.,* an error involving the use of a definite plural noun phrase instead of an indefinite one would typically be conceptualised as a superfluous *the*).

2.3.2. Linguistic category classification

As an alternative to FITIKIDES' surface structure taxonomy, BURT & KIPARSKY's collection of errors (1972) presents an organisation according to more linguistically motivated major categories (*see* Appendix A.1 for more details):

1. Clause skeleton
2. Auxiliary system
3. Passive
4. Temporal conjunctions
5. Sentential complements
6. Psychological predicates

Despite completely different starting points, the two taxonomies actually comprise many of the same more fine-grained categories and individual types of error (*e.g.,* infinitive for gerund after preposition, misuse of *isn't it?* or missing *be* from passive construction), the main difference in substance being perhaps FITIKIDES' attention to lexis as compared to BURT & KIPARSKY's focus on syntax, which is reflected in the way the material has been organised in each.

A more complete taxonomy of errors in written texts is provided by JAMES (1998, Ch. 5, *q.v.* for details), as briefly summarised in the following:

1. Substance errors
    a) Mechanical errors
        i. Punctuation

ii. Typographic (keyboard-related)
iii. Dyslexic
iv. Confusibles (*anus/onus, cords/chords*)

b) Misspellings proper

i. Mispronunciation
ii. Written misencoding (inter-/intralingual)

2. Text errors

a) Lexical errors

i. Formal errors of lexis

α′ Formal misselection (malapropism, deceptive cognate)
β′ Misformation (borrowing, coinage, calque)
γ′ Distortion

ii. Semantic errors in lexis

α′ Confusion of sense relation (hypo-/hypernym, quasi-synonym)
β′ Collocational errors, verbosity

b) Morphological errors

i. (noun, verb, adjective, adverb)

c) Syntactic errors

i. Phrase structure

α′ Misselection or misordering of constituents (modifier, head, qualifier)

ii. Clause

α′ Combination of phrases

iii. Sentence

α′ Blends
β′ Coordination
γ′ Subordination

iv. Intersentence (cohesion)

α′ Reference
β′ Substitution
γ′ Ellipsis
δ′ Conjunctions (over-/underuse, misselection, misplacement)
ε′ Lexical

3. Discourse errors

a) Coherence errors

i. Topical

ii. Relational
iii. Sequential

b) Pragmatic errors[6]

i. Taboos
ii. Size of the imposition
iii. Values
iv. Power and social distance

Extending the traditional division of grammar into morphology and syntax, this taxonomy systematically and exhaustively lists the different levels at which an error can occur — starting with single letters and punctuation marks, going through words, phrases, sentences and paragraphs, ending with the semantic content of an entire text — and further characterises the types of error at each level according to applicable criteria such as part of speech, part of phrase or origin of the error; the hierarchy of constituent parts is more or less given, but there are still many ways to subdivide the errors at each level (*cf.* Politzer & Ramirez 1973 for an example of categories closer to those found in traditional grammars).

2.3.3. Extent and context

In general, local errors such as spelling mistakes are easier to detect and correct reliably than global ones such as discourse errors (which have, at least partly for this reason and perhaps regrettably, been somewhat neglected in practical corpus annotation work, but that is not our concern at this stage). However, as pointed out by Lennon, the degree of localness depends not only on the 'extent' of the error, but also on 'the breadth of context [...] criterial for whether error has occurred', its 'domain', which may vary from a single morpheme to, in the extreme case, extralinguistic context (1991). In other words, even a trivial typographic error may require a large amount of contextual information to become apparent, in which case detecting it may not at all be a simple task despite its minute extent.

There is thus no absolute correlation between the level at which an error belongs in the linguistic hierarchy and the ease with which it can be handled. For this purpose, a more appropriate linguistically motivated hierarchy might be based on the level of descriptive detail needed to detect that something is amiss, originally proposed to explain degrees of grammaticality (Chomsky 1961): at the lowest level, any sequence of English words is acceptable; at the next level, words are divided into parts of speech and their combination

[6]) Whereas pragmatic considerations, for instance with respect to the right level of formality in a given context, are indeed relevant in written communication, pragmatic errors are perhaps more common in face-to-face interactions; in particular, the example given for 'size of the imposition' is the idea that it will be more acceptable to ask a random stranger for a cigarette in places where tobacco products are easy to come by and relatively inexpensive.

must conform to certain syntactic rules, which would exclude something like *the *cater ate my gymsuit;* at a higher level, verbs may be subdivided into 'pure transitives, those with inanimate objects, etc.', nouns subdivided similarly, and the rules refined accordingly, at which point sufficient information would be available for an anomaly like *the *khat ate my gymsuit* to be detected. The practicality of an error classification scheme along these lines, as well as the adequacy of the resulting categories, remains to be established.

2.3.4. Taxonomies used for error annotation

A problem with extant taxonomies known from language manuals and the linguistic literature, be they organised according to surface structure or based on linguistic categories, is that it is often not clear to which category a given error should be assigned; for instance, *draftsman* for *draughtsman* could reasonably be classified as a phonetic spelling (written misencoding), a confusion between homonyms or influence from North-American or Australian spelling conventions. The original author would typically be able to indicate the cause of an error, but this approach is clearly not feasible for corpus annotation, and there would in any case remain instances where not even the author is able to tell with certainty, — or the correct explanation might involve a combination of two or several causes. A more practical solution is to construct the set of error types in such a way as to eliminate, or at least minimise the amount of, overlap between categories; one way of achieving this is to use descriptive rather than explanatory categories.

A related problem appears when a given error can be corrected in more than one way, such as **friends his* for *his friends* (word order error) or *friends of his* (missing preposition). In this particular case, having a category for 'incorrect possessive formation' would circumvent the issue, but specific guidelines are typically needed to deal with such cases in a consistent manner. Another option would be to assign a given error to multiple categories, but this is usually not considered, not least because the annotation task becomes even harder if one has to ensure that all error categories corresponding to equally good or at least plausible corrections are included consistently.

Most error taxonomies devised for corpus annotation combine surface structure and linguistic categories, as for instance the scheme used in the CLC (*see* Fig. 2.1): the majority of the error types indicate both the part of speech involved and the general category of error, expressed either in terms of word-level surface structure modification (missing word, unnecessary word, etc.) or in terms of linguistic category (agreement, derivational morphology, etc.); additional error types are used when multiple parts of speech are involved (*e.g.,* word order) or when the part of speech is found to be irrelevant (*e.g.,* spelling or register). Errors at the word level and below are categorised along lines similar to James's linguistic taxonomy, whereas syntactic errors are dealt with in terms of surface structure. The error types are largely non-overlapping and self-explanatory, but there are examples of

	Unspecified part of speech	Verb	Noun	Adjective	Adverb	Preposition	Conjunction	Quantifier	Pronoun (anaphor)	Determiner	Punctuation
Wrong form used		FV	FN	FJ	FY			FQ	FA	FD	
Something missing	M	MV	MN	MJ	MY	MT	MC	MQ	MA	MD	MP
Word or phrase needs replacing	R	RV	RN	RJ	RY	RT	RC	RQ	RA	RD	RP
Word or phrase is unnecessary	U	UV	UN	UJ	UY	UT	UC	UQ	UA	UD	UP
Word is wrongly inflected		IV	IN	IJ	IY	DT		IQ	IA	DI	
Word is wrongly derived		DV	DN	DJ	DY			DQ	DA	DD	
Verb is in the wrong tense		TV									
Countability error			CN					CQ		CD	
Agreement error	AG	AGV	AGN					AGQ	AGA	AGD	
Wrong spelling	S										
Spelling confusion	SX										
US spelling	SA										
Collocation/tautology error	CL										
Register error	L										
Negative formation error	X										
Complex error	CE										
Idiom error	ID										
Argument structure error	AS										
Word order	W										

FIGURE 2.1. Systematic overview of the error codes in the CLC (NICHOLLS 2007).

somewhat problematic and probably not very useful distinctions such as the one between determiners and quantifiers, as well as a set of arbitrary rules such as the one saying that confusion between two words of the same part of speech (*e.g., flower* and *flour*) should be regarded not as an instance of spelling confusables (SX) but instead as a replacement error (RN in the case of nouns). There used to be a category for deceptive cognates ('false friends'), but this one has now been removed since the annotators cannot be expected to recognise cognates reliably for a large number of different languages; this additional more explanatory category also overlapped with others, in SCHOLFIELD's words 'a sure sign of a faulty scale' (1995). Assuming that an error can be detected, the annotator should be able to assign it to a category without knowing the author's mother tongue or what was going on in his head when the error was committed; the classification should be based entirely on what can be observed from the text.

A good review of existing error annotation taxonomies is provided by DÍAZ-NEGRILLO & FERNÁNDEZ-DOMÍNGUEZ (2006). Compared to the CLC taxonomy, others used for annotating larger corpora such as the International Corpus of Learner English (ICLE) at The Université catholique de Louvain (*see* Appendix A.4) or the Japanese Standard Speaking Test (SST) corpus (*see* Appendix A.3), are mainly organised according to a linguistic classification, including categories for complementation of noun and voice of verb; on the other

hand, the main difference seems to be the formal organisational hierarchy of error types rather than the distinctions made between different types of error, and surface structure categories cannot easily be avoided altogether given that word order errors in particular are difficult to define in other terms. The FreeText taxonomy (*see* Appendix A.2) is sometimes characterised as three-tiered, but it would probably be more accurate to say that it is two-dimensional, combining an error category with a part-of-speech category just like in the CLC, but with two potential improvements, namely additional error categories and consistent encoding of part of speech.

2.3.5. Other approaches

In addition to the largely descriptive taxonomies described in previous sections, other classifications of errors have been developed with the aim of diagnosing the origin of an error, explaining it from a language-learning point of view, or evaluating how serious it is, for instance in a communicative perspective. Most of these dimensions fall outside the scope of this thesis, but it can sometimes be difficult to keep description and explanation completely separate, as we have already seen in the case of cognates and misspellings which happen to result in a valid word or an alternative spelling which is acceptable in another geographical variety of the language. A radical way of avoiding this issue is not to classify errors at all, but provide only a reconstruction, as suggested by FITZPATRICK & SEEGMILLER (2004). A more ambitious approach was chosen by the developers of FALKO (Fehlerannotiertes Lernerkorpus des Deutschen), who chose to encode description and explanation at different levels (LÜDELING *& al.* 2008).

Something that seems to be missing from almost all approaches to error annotation is a perspective of how it can interact with other means of corpus analysis. For instance, adding part-of-speech tags is all very well, but this is perhaps something that could be done automatically, especially if a corrected version of the erroneous passage is provided anyway; conversely, information latent in the error annotation could be helpful for parsing syntactically incorrect sentences successfully. More generally, when multiple types or levels of annotation are to be added to the same text, taking advantage of interdependencies between them to avoid encoding essentially the same information several times can greatly reduce the amount of work involved, a principle referred to by PIENEMANN as 'economical exploitation' of pre-existing annotation (1992).

2.4. Error detection

Precursors to computer-based writing tools emerged in the late 1950s in the form of methods for automatic correction of transmission errors, such as the ones introduced by optical character recognition, and approximate or 'elastic' string matching algorithms, typically

needed for personal names that are not always written down correctly; many academic studies on the general problem of string matching and correction were to follow; and perhaps the first spelling checker to become widely distributed, *spell*,[7] has been around since 1971 (PETERSON 1980).

KUKICH's comprehensive literature review (1992) identified three distinct subtasks: non-word error detection (*e.g., teh*), isolated word error correction (*e.g., teh* corrected to *the*) and context-dependent detection and correction, predominantly exemplified by errors owing to orthographic similarity between two words (*e.g., too* in *the too words* detected as erroneous and corrected to *two*), but this category also covers other types of inappropriate use of individually correct words.

2.4.1. Non-word detection

For a rather isolating language like English, with limited derivational and inflectional morphology, non-words can be identified fairly reliably by looking up word forms (sequences of letters delimited by punctuation marks or white space) in a lexicon. For this to work well, the lexicon should be of limited size and adapted to the writer: it should ideally be not only sufficiently comprehensive to include any word used in a text and thus avoid flagging correct words as misspellings, but also sufficiently limited not to include infrequent words, technical terms outside the author's field or variant forms belonging to other dialects given that the chance of an accidental match between a misspelling of one word and the correct spelling of another increases with the size of the lexicon. The standard approach is to include a medium-size dictionary of fairly common words and give the user the ability to add words to a personal word list.

After half a century of existence, non-word detection is often regarded as a solved problem, but there is still room for improvement, for instance when it comes to proper names or handling of white space (or absence thereof) in noun compounds, not to mention consistency checking in cases where more than one spelling is acceptable (*e.g., -ise/-ize* verb suffixes).

2.4.2. Isolated word error correction

Once a non-word has been detected, a system can either simply flag it as misspelt or try to come up with one or more suggestions as to what the intended word might be. In DAMERAU's data, 80 per cent of all misspellings could be explained as resulting from a single insertion of a letter, removal of a letter, replacement of one letter by another or transposition of two adjacent letters, which are 'the errors one would expect as a result of misreading,

[7]) Now known as *International Ispell*, unrelated to Unix *spell*.

hitting a key twice, or letting the eye move faster than the hand' (1964). Already the original *spell* hypothesised corrections based on these transformations (PETERSON 1980), an approach which can be expected to work well for purely mechanical errors, but less so for cognitive errors unless they happen to have the same characteristics as mechanical errors (*e.g.*, **decieve*). Levenshtein later introduced the concept of edit distance, defined as the minimum number of atomic operations needed to transform, in our case, a misspelt word into its correct counterpart; unlike Damerau, he only considered insertions, removals and replacements, but the resulting Levenshtein distance can easily be generalised to include transpositions as a fourth category, which gives the Damerau–Levenshtein distance. Other edit distances have been developed since, and methods have been devised to deal with the problem of finding a correction in the case of there being no word in the dictionary at distance 1, and to take into account that certain letters are more often confused, left out or added, than others (BRILL & MOORE 2000; AHMAD & KONDRAK 2005); in addition, completely different techniques have appeared (see KUKICH 1992 for an overview), including ones based on the use of a similarity key (*e.g.*, Soundex, *cf.* RUSSELL 1918) or explicit rules. More recent developments include automatic derivation of rules (MANGU & BRILL 1997) and the use of complementary sets of rules to handle errors arising from different causes (DEOROWICZ & CIURA 2005), an idea already presented as pre-existing by PETERSON (1980).

2.4.3. Context-sensitive spelling errors

The methods mentioned so far are unable to detect real-word errors, a category found to constitute between 25 and 40 per cent of the total number of errors in two empirical studies (KUKICH 1992). Classical approaches include hand-written rules, grammar-based techniques and *n*-gram statistics (see *ibid.* and DICKINSON 2006). *n*-grams are usually made up of either words/lemmata or parts of speech, but the two can of course be combined, or another variation on the theme can be devised, like HUANG & POWERS's reduction of content words to affixes, combined with complete function words (2001). Lexical bi- or trigrams can potentially handle local errors like **piece prize* for *peace prize*, but *n*-grams are not suitable for modelling slightly less immediate contextual effects such as the likelihood of *desert* being a misspelling of *dessert* in the neighbourhood of *crème brûlée* and *crêpes Suzette*, with perhaps dozens of words intervening, a problem which can be solved by using surrounding words individually as separate lexical features; the two methods are largely complementary and can be combined, as suggested by GOLDING (1995), amongst others. In a similar vein, HIRST & BUDANITSKY presented a method for detection and correction of semantic anomalies (*e.g., it is my sincere *hole/hope*) by considering local and global context and looking for orthographically related and contextually plausible alternatives to a contextually anomalous word. At the level between part-of-speech tags and complete parses, SJÖBERGH suggested to use *n*-grams of phrase labels (noun phrases, verb chains,

prepositional phrases, etc.), potentially combined with actual words/lemmata; rare phrase sequences would be indicative of an error (2005).

2.4.4. Grammar checking

Context-dependent spelling errors are similar to syntactic and stylistic errors in that more than a single word must be considered in order for them to be identified, and many of them are typically handled by 'grammar checkers' rather than 'spelling checkers'. Perhaps the first widely available tool to deal with syntactic and stylistic issues was the Writer's Workbench (MACDONALD *& al.* 1982); during the 1980s, several tools were developed and commercialised as separate products, and more advanced syntactic checking was gradually added; in 1992, Microsoft Word and WordPerfect both integrated grammar checkers as part of the word processor, which has since become ubiquitous and thus made grammar checking readily available to the general public.

Automatic writing aids have the merit of being able to point out errors that the writer might not have spotted if left to his own devices, but an author is well advised not to rely on a grammar checker; currently popular commercial systems have not only low recall, which means that many common errors will remain undetected, but also limited precision, which means that correct language will sometimes be flagged as questionable or incorrect, leading to errors being *introduced*, induced by the grammar checker, if the user puts too much trust in the suggestions provided by the machine. The imperfect nature of such tools is not only problematic for poor spellers and people who find it difficult to express themselves well in writing; competent writers can fall under the computer's spell by being lulled into a false sense of security and actually produce texts containing more errors than they would without mechanical assistance (GALLETTA *& al.* 2005).

It will of course not come as a surprise to a computational linguist that grammar checkers are imperfect and not on their own sufficient to guarantee correct and good use of language, and also early users and reviewers seem to have been aware of the limitations:

> Just as a paint program won't make you an artist, a grammar and style checker won't make you a professional writer.
>
> — EGLOWSTEIN 1991

Exactly how well or how badly different programs perform is difficult to quantify since the set of errors, and even types of error, they should catch is open-ended. Some studies have nevertheless been performed, including two focusing on errors committed by university students, KOHUT & GORMAN using business students' writing samples (1995) and KIES using sentences exemplifying the 20 most common usage errors found in a sample of 3,000 essays (2008). The results are not directly comparable given that the two studies use different sets of errors for testing, but the results are similar: the best system in each study

has an accuracy of 50 per cent or just below, and the other good systems work with an accuracy of between 30 and 40 per cent. KOHUT & GORMAN tested software available in the early/mid 1990s, whereas KIES covered the period 1997–2007 and explicitly mentioned that no actual progress had taken place during that time. These results are consistent with Bruce Wampler's (the original author of Grammatík, which became part of WordPerfect) claim that 'Microsoft has decided that its [...] grammar checker is "good enough" and has stopped significant work on improvement' (KIES 2008). The allegation that the current/traditional grammar checker in Word should have been somewhat abandoned was corroborated by AIKAWA from Microsoft Research, who explained that the grammar-based approach to detecting well-known usage errors could not be adapted to other languages without requiring an amount of work essentially amounting to developing a new grammar checker from scratch for each language, and that other approaches were therefore currently under investigation (2008).

Owing to the stagnation of integrated spelling and grammar checkers, separate programs, now including web-based ones, have seen a certain resurgence lately with for instance WhiteSmoke[8], Ginger[9] and StyleWriter[10].

2.4.5. Children and foreigners

As mentioned in the previous section, grammar/style checkers typically assume that the user already knows the language well; they were originally developed as an aid for adult, educated, native speakers of English to detect punctuation errors, confusions between similar words, split infinitives, wordiness, excessive use of the passive voice and other constructions regarded either as potentially stigmatising errors of grammar or as suboptimal according to advocates of 'clear writing'. The types of error more frequently committed by children (HASHEMI, COOPER & ANDERSSON 2003) or by non-native speakers will therefore often not be covered. LIOU, who developed an early system for detecting errors in Taiwanese students' writing, found that contemporary commercial grammar checkers detected (or misdetected) stylistic errors whilst overlooking significant syntactic errors like **having listening something* for *having listened to something* (1991). Her system instead used specific rules to detect common learner errors at the sentence level (*e.g.*, subject–verb disagreement), at the noun phrase level (*e.g., one of* followed by singular) and at the verb phrase level (*e.g.*, unbalanced coordination like **creates science and lived happily ever after*), as well as capitalisation errors. Many of the errors can of course be committed by native speakers as errors of inattention, but they will not usually be very frequent, and a native speaker is less likely to be led astray by incorrect diagnoses.

[8]) http://www.whitesmoke.com/

[9]) http://www.gingersoftware.com/

[10]) http://www.stylewriter-usa.com/

More recent approaches to error correction developed with learners in mind include parsing with mal-rules (BENDER *& al.* 2004; FOSTER & VOGEL 2004), parsing of different hypothesised permutations of the original string (VLUGTER, HAM & KNOTT 2006) and statistical methods (CHODOROW & LEACOCK 2000). Several methods will be considered more in detail in the following chapters.

Generally available tools specifically aimed at non-native writers include ProofWriter[11] from Educational Testing Service and ESL Assistant[12] from Microsoft Research.

2.5. Conclusion

We have discussed the concept of error and proposed as a practical definition linguistic productions that deviate from what one would expect in a given situation, including semantic mismatch between what was said and what was intended only insofar as this is observable in the sense of being truly obvious from the situation. Furthermore, errors which are in principle possible to detect may be obscured, for instance by the lack of contextual or situational information or when sentences are evaluated in isolation and not as part of a larger unit.

Real linguistic errors have been collected, classified and analysed in different ways for different purposes; and a series of grammar checkers and other tools for writers have been developed and become widely used. Unfortunately, though, with the exception of certain tools aimed at foreign learners, much of the existing software seems to have been developed without due consideration of the types of error typically committed and therefore useful to detect.

Given the unavailability of a native-speaker error corpus, the errors used as training and test data for the work reported in this thesis are all committed by non-native speakers, more precisely during language examinations and recorded in the Cambridge Learner Corpus. It would seem plausible that errors committed by adult native speakers should be a subset of those committed by foreign learners, and somewhat similar to the ones committed by advanced learners under similar conditions, although this is difficult to verify, and the proportions of different types of error are likely to be different.

[11]) https://proofwriter.ets.org/

[12]) http://www.eslassistant.com/

Chapter 3.

Bits and pieces

We have already alluded to the use of corpora as evidence of correct as well as incorrect constructions, which presupposes that the corpus exists in an appropriately exploitable form. One metaphor for a corpus is a paper tape on which the words are written one after another, but such a representation (in the sense of an electronic equivalent) is not adequate for many purposes: at the very least, one would probably want to know where the title of an article ends and the first sentence begins in the absence of punctuation, even if titles are treated as ordinary sentences; if a corpus contains texts belonging to different genres, or both written and spoken material, the different categories must be separable somehow; and one might want to add syntactic annotation, develop indexes for efficient searching, and so on. In this chapter, we look at how the current standard of encoding corpora in an xml format has emerged, and how information provided by a parser can be added to a corpus. The importance of parts of speech and grammatical relations thus added for the purposes of grammatical error detection will become apparent in the following chapters.

3.1. Encoding

Almost the entire Brown corpus was 'competently and cheerfully punched' by a single operator (Kučera 1967, p. ix), using tried and tested 80-column ibm punched cards with rectangular holes first introduced in 1928 (Kistermann 1991). The punching scheme for encoding letters in addition to decimal digits had been in place at least since 1934 (*ibid.*), but the character set remained extremely limited: no distinction was made between upper and lower case, and only a dozen punctuation marks were available. This limitation was overcome by adopting and adapting a notation system devised a few years earlier by Newman, Swanson & Knowlton for the encoding of us patents (1959). Fundamental characteristics include the use of a prefixed asterisk to indicate upper case and special codes for formatting information and document structure (*see* Appendix b for details). The

encoding of Roman numerals as Arabic numerals surrounded by special codes and the systematic distinction between abbreviation point and full stop (both will appear in sequence after a sentence-final abbreviation) are examples of neat concepts which have not generally been imitated by later corpora, whereas the lack of discrimination between ampersand and plus sign is somewhat surprising. Several versions of the corpus have been produced over the years (Francis & Kučera 1964), partly to take advantage of technical advances such as the ability to encode lower-case letters directly, partly to make the corpus more immediately useful for particular applications (*e.g.*, a version without punctuation marks for word frequency tabulation). The tagged version was encoded in a fixed-width format with 30 characters (or 'columns' in punched-card terminology) reserved for the actual word or punctuation mark and 10 characters for the part-of-speech tag.

The lob Corpus (Johansson, Leech & Goodluck 1978) was encoded using the less restrictive ascii character set from the onset, which reduced the need for frequently used compound symbols by allowing a larger number of characters to be encoded directly, and also made it possible to devise a more readable and mnemonic coding scheme. New codes were added to represent such concepts as the start of a paragraph or a list of items, and provision was made for European characters including Old English letters, Cyrillic in addition to Greek, and a fairly comprehensive set of accents and other diacritical marks in addition to the diæresis. Most of the modifications can be characterised either as improvements made possible by a larger underlying character repertoire or as extensions added to handle characters not typically found in us patents; the fundamental model of a text format enhanced with a number of specific codes remained. The tagged version was encoded in two different ways, 'vertically' with one word per line akin to the tagged version of the Brown corpus, and 'horizontally' with the text flowing normally, the part-of-speech tag being added immediately after each word, separated by an underscore.

When the bnc was being designed, there was a desire to record bibliographic information and other metadata within the corpus itself; previous corpora had to a certain extent documented the material included in a separate reference manual and provided a limited way of extracting sub-corpora by grouping texts of the same genre together, an approach which was seen as impractical for the compilers and inadequate for the users of a corpus comprising thousands of texts classified according to media, level, region and other criteria, and not simply assigned to a monolithic genre. Another major concern was the inclusion of part-of-speech tags as an integral part of the main corpus and not 'only' in a derived version, which for instance meant that *cannot* could not be represented as something like *can_VM0 not_XX0* after tagging since it would then be indistinguishable from *can not*. Burnard provides some information on the development of the bnc and its encoding format (1999).

As a solution to these and other issues, and to assure its longevity, it was decided to encode the corpus in an format based on the Standard Generalized Markup Language (sgml), with mark-up providing structural information like word and sentence boundaries as well

as a framework for indicating meta-information. In more technical terms, the encoding scheme to be used for the BNC, called the Corpus Document Interchange Format (CDIF), was defined as an application of SGML. The more general Text Encoding Initiative (TEI) guidelines were developed in symbiosis with the BNC-specific CDIF and have since been adopted by hundreds of corpora and text collections.

Like earlier corpora, the original version of the BNC was encoded using a limited character set, this time a subset of ASCII,[13] but SGML provided a standard means of representing additional letters and symbols through character references, and almost all the ones required in the BNC had already been standardised and were listed in an appendix to the standard (GOLDFARB 1990).

From the outset, it was envisaged that a large proportion of the written corpus could be constructed from sources which already existed in electronic form, but the availability of such texts turned out to be smaller than expected, and the typesetting tapes that did exist were often too difficult to convert, which meant that much of it had to be scanned or typed instead. Each method introduces errors, but of different kinds: conversion of electronic sources may introduce errors if the mapping is not accurately defined, which is particularly likely to go unnoticed for somewhat rare characters and probably explains the substitution of *Zu\3rich* for *Zürich*; systems for optical character recognition easily confuse visually similar characters and are likely to deserve the blame for *lst* instead of *1st* and | instead of *I;* typists may introduce keyboard-related mistakes such as *tele[hone* for *telephone* (given that *[* and *p* occupy adjacent positions on an English keyboard), but also introduce misspellings or unconsciously fail to reproduce real or apparent errors in the original, and such errors are both practically inevitable and difficult to detect (*cf.* MITTON, HARDCASTLE & PEDLER 2007 for a discussion of errors in the BNC, be they faithfully copied from the original source or introduced as part of the corpus development).

With the rise of the World Wide Web and the success of its HyperText Markup Language (HTML) appeared the idea of 'an extremely simple dialect of SGML' (BRAY & SPERBERG-MCQUEEN 1996) which could be processed by much simpler parsers than the ones needed to deal with the full language. At that point, the twin standards Unicode & ISO/IEC 10646 defined a universal character set, potentially putting an end to the proliferation of character sets for different languages and purposes, and this was chosen as the document character set[14] (which is left unspecified by SGML itself) for the simplified dialect. An important guiding principle was that '[t]erseness [should be regarded as] of minimal importance' (*ibid.*), which allowed a set of mark-up abbreviation techniques to be eliminated. The resulting

[13]) The apparent intention is to restrict the repertoire to ISO 646 INV, but this is not enforced by the SGML header, and a few occurrences of characters outside this set have crept in.

[14]) This refers to the 'abstract' character set, the repertoire of characters that can occur in a document; a 'concrete' realisation of a document can use any 'external' character encoding to define a mapping from concrete bytes to abstract characters; character references always refer to abstract characters irrespective of the external encoding, which allows characters not covered by the encoding to be used.

Extensible Markup Language (XML) has become much more widespread than its progenitor, and the mark-up scheme used in the third edition of the BNC (BURNARD 2007) was defined as an application of XML. At the same time, named character entities were replaced by Unicode characters encoded in UTF-8, and a number of corrections and improvements were made at different levels (*cf.* PRYTZ 2007).

BURNARD estimated that not using SGML tag minimisation in the BNC 'would have more than doubled the size of the corpus' (1999), which appears to be a fairly accurate characterisation of what happened when this facility had to be abandoned with the move to XML; actually, a superficial examination shows that the corpus size has trebled, but this is partly due to the addition of lemmatised forms and a coarse part-of-speech category. The difference in file size can at least partly be offset by more efficient compression achieved for the more repetitive XML format, though, and the availability of XML tools means that the corpus is more readily exploitable in this format.

Actual characters instead of character entities generally represent an improvement as well, given that most tools are now able to handle Unicode, and a couple of meaningless or unreliable distinctions such as the ones between *&half* and *&frac12* for the fraction ½ or between *&ft* and *&prime* for the prime (′) will not be missed. Some arguably useful distinctions have also disappeared, however, including the one between a regular hyphen and one appearing at the end of a line, and thus likely not to be part of the word's normal orthography.

In general, the actual conversion from SGML to XML should be fairly straightforward given adequate tools and sufficient expertise, so it is somewhat disheartening to note that the third edition of the BNC contains errors not present in the first edition and apparently introduced as part of the conversion process. Trivial examples include the substitution of < (encoded as *&lt*) and > (*&gt*) for ≪ (*&Lt*) and ≫ (*&Gt*), not to mention truly bizarre cases like the extraneous semicolon and space introduced into $(FR)' = R'F'$ to yield $(FR)' = R$; $'F'$ (the corresponding SGML mark-up being *(FR)&prime = R′F&prime*). The errors mentioned here are arguably not of great importance as far as using the corpus for linguistic enquiry is concerned and quite likely quantitatively less important than the inaccuracies introduced by the original conversion/scanning/typing, although the very existence of such unexpected errors somewhat undermines the claim that formats based on SGML or XML are a solution to the perennial problem of viable long-term data storage. Quantitatively more important are the thousands of empty elements introduced as a result of the somewhat unsuccessful conversion of formulæ marked up as *<w UNC>&formula* in the SGML edition to *<w c5="unc" hw=" " pos="UNC> </w><gap desc="formula"/>* in the XML edition, by which the mark-up defining the formula as a word of unknown part of speech has been transferred to a spurious space in front of it. These and other errors have been reported to and acknowledged by Lou Burnard, who has suggested that they will be corrected in a future version of the corpus to the extent possible without extensive manual intervention.

3.2. Annotation

In their raw form, corpora may provide the necessary material for lexicon compilation, different types of empirical investigations of language use such as statistical analysis of word and letter frequencies, or automatic generation of concordance lists for particular words of interest to researchers or language workers (*cf.* LUHN's keyword-in-context (KWIC) index, 1960). A small amount of metadata characterising the individual pieces of text in terms of their origin, nature and so forth furthermore allows to derive, for instance, information about lexical variation across genres or depending on the date of publication.

En revanche, someone who would like to study something like the usage of the noun *can* would have to go through an overwhelming number of occurrences of the homographic auxiliary if an unannotated corpus were used as a source of concordances, whereas extracting instances of such a simple concept as a past participle followed by a preposition would not be possible at all. For a corpus to be useful in scenarios such as these, each word must be assigned its part of speech. The idea of mechanical part-of-speech assignment had already been put into practice by the time the Brown corpus was first published (KLEIN & SIMMONS 1963); and such a tool, Taggit (GREENE & RUBIN 1971), was subsequently put into use during the development of a part-of-speech-tagged version of the corpus, but the accuracy was only about 77 per cent (MITKOV 2003), so the necessary post-editing was 'a long and arduous process, extending over several years' (FRANCIS & KUČERA 1964) which was not completed until 1979. In comparison, the Constituent-Likelihood Automatic Word-Tagging System (CLAWS), used to tag the LOB corpus, was able to assign the correct tag for 96–97 per cent of the words (GARSIDE 1987), at least in part thanks to information derived from the tagged version of the Brown corpus, including tag sequence probabilities used for disambiguation. For the British National Corpus (BNC), manual post-editing was found to be impractical; part-of-speech tags were assigned by a newer version of CLAWS with an accuracy estimated to be over 98.8 per cent (at the expense of leaving almost 4% of the words with a portmanteau tag, ambiguous between two parts of speech). Only Benjamin Zephaniah's poetry was tagged manually, since the tagger was 'not familiar with Jamaican Creole' (LEECH & SMITH 2000), an extract from which is included below:

> Dis poetry is not afraid of going ina book
> Still dis poetry need ears fe hear an eyes fe hav a look
> Dis poetry is Verbal Riddim, no big words involved
> An if I hav a problem de riddim gets it solved,
>
> — BNC file F9M, from ZEPHANIAH 1992

Corpora with manual or manually checked syntactic annotation range from the detailed and thoroughly checked analyses of 128 thousand words in the Surface and Underlying Structural Analysis of Naturalistic English (SUSANNE) corpus (SAMPSON 1995) to the more skeletal bracketing of *c.* 3 million words in the Penn Treebank (MARCUS, SANTORINI &

Marcinkiewicz 1993; Marcus *& al.* 1994), whereas the linguistic annotation in larger corpora is usually limited to part-of-speech assignment and lemmatisation. Other types of annotation being largely restricted to specialised corpora, the Manually Annotated Sub-Corpus (masc) of the American National Corpus, which is currently under development, aims to fill

> the critical need for sharable, reusable annotated resources with rich linguistic annotations [...] including WordNet senses and FrameNet frames and frame elements [...as well as] shallow parses and named entities
>
> — Ide *& al.* 2008

The examples given so far show that annotation may render a corpus more imminently exploitable; it represents a valuable 'enrichment' (Leech 2005). This view is challenged by Sinclair, who maintains that '[a]nnotation loses information' by leading the researcher to take provided parts of speech, for instance, for granted instead of re-examining the actual words (2004); this particular criticism is related to the idea that 'a mark-up system can easily become a prison' (Scott & Tribble 2006, p. 32): pre-existing corpus annotation, useful though it may be, reflects a certain way of looking at language, with a limited number of categories and a given level of granularity, which will not always be appropriate for the task at hand. To avoid this problem, and for the sake of scalability, Sinclair argues that '[a]nalysis should be restricted to what the machine can do without human checking, or intervention' (1992), non-automatic annotation of generic corpora being a 'very flawed activity', though this comment does not extend to corpora 'designed and built for a pre-determined application' (2004).

The remainder of this this chapter is devoted to practical issues of adding annotation to a corpus.

3.3. Error-coded and parsed version of the CLC

After manual error-coding, an erroneous sentence from the Cambridge Learner Corpus might end up as the following:

(8) *Then* <RD>*some|a*</RD> <SX>*though|thought*</SX>
<IV>*occured|occurred*</IV> *to me.*

The original text as written by the candidate can be extracted, as well as a corrected version constructed using the corrections provided by the annotator:

(9) a. *Then some though occured to me.*
b. *Then a thought occurred to me.*

In addition, the error codes tell us that this particular sentence contains three errors: a determiner replacement (misselection) error, an orthographic confusion between two similarly spelt words, and an incorrect verb inflection.

The error-coding provides explicit information about errors, but is mostly limited to the lexical level, not in the sense that syntactic errors are ignored, but they are difficult to analyse automatically since parts of speech and syntactic relations between words are not indicated. One way of adding syntactic information is to use a parser, both on the original and on the corrected version of each sentence; the CLC has been parsed in this way using the RASP system (BRISCOE, CARROLL & WATSON 2006), which provides tokenisation, part-of-speech tagging and lemmatisation:

(10) a.

RR	DD	RR	VVN	II	PPIO1	.
Then	*some*	*though*	*occur+ed*	*to*	*I+*	.

b.

RR	AT1	NN1	VVD	II	PPIO1	.
Then	*a*	*thought*	*occur+ed*	*to*	*I+*	.

In this example, *occured* in the incorrect version of the sentence has been mistagged as a perfect participle (VVN) instead of a preterite (VVD), whereas the spelling error itself has disappeared through lemmatisation.

Furthermore, the RASP parser provides a parsing tree (*see* Fig. 3.1) and grammatical relations (*see* Fig. 3.2). Complete XML code for the example sentence can be found in Appendix D.1.

The RASP system is a domain-independent, robust parsing system for English which is free for research purposes. It was, in common with other extant publically available parsers, designed for plain-text input and has only limited ability to handle XML-style mark-up natively. For the purposes of parsing the CLC, Rebecca Watson wrote corpus-specific code to extract the text to be parsed and construct an XML document incorporating the original error annotation as well as the added syntactic annotation. A newer version of the parsed CLC has been indexed to make it more useful and usable for linguistic research (GRAM & BUTTERY 2009).

Issues directly related to parsing the CLC will not be discussed further here; however, many of the issues are the same for different types of corpora, so much of the following discussion on how best to parse the BNC also applies to the CLC and other corpora. As for the choice of parser, it can be argued that a non-lexicalised parser such as RASP can be expected to give more reasonable analyses of sentences containing misspellings and other errors, which is obviously important in the case of a learner corpus, although this has not been tested experimentally; we chose the same parser for the BNC to minimise the risk of diverging analyses of the same or related constructions in the two corpora, but several other parsers are able to produce grammatical relations that may well be sufficiently similar to the ones provided by RASP for this not to be an issue in practice.

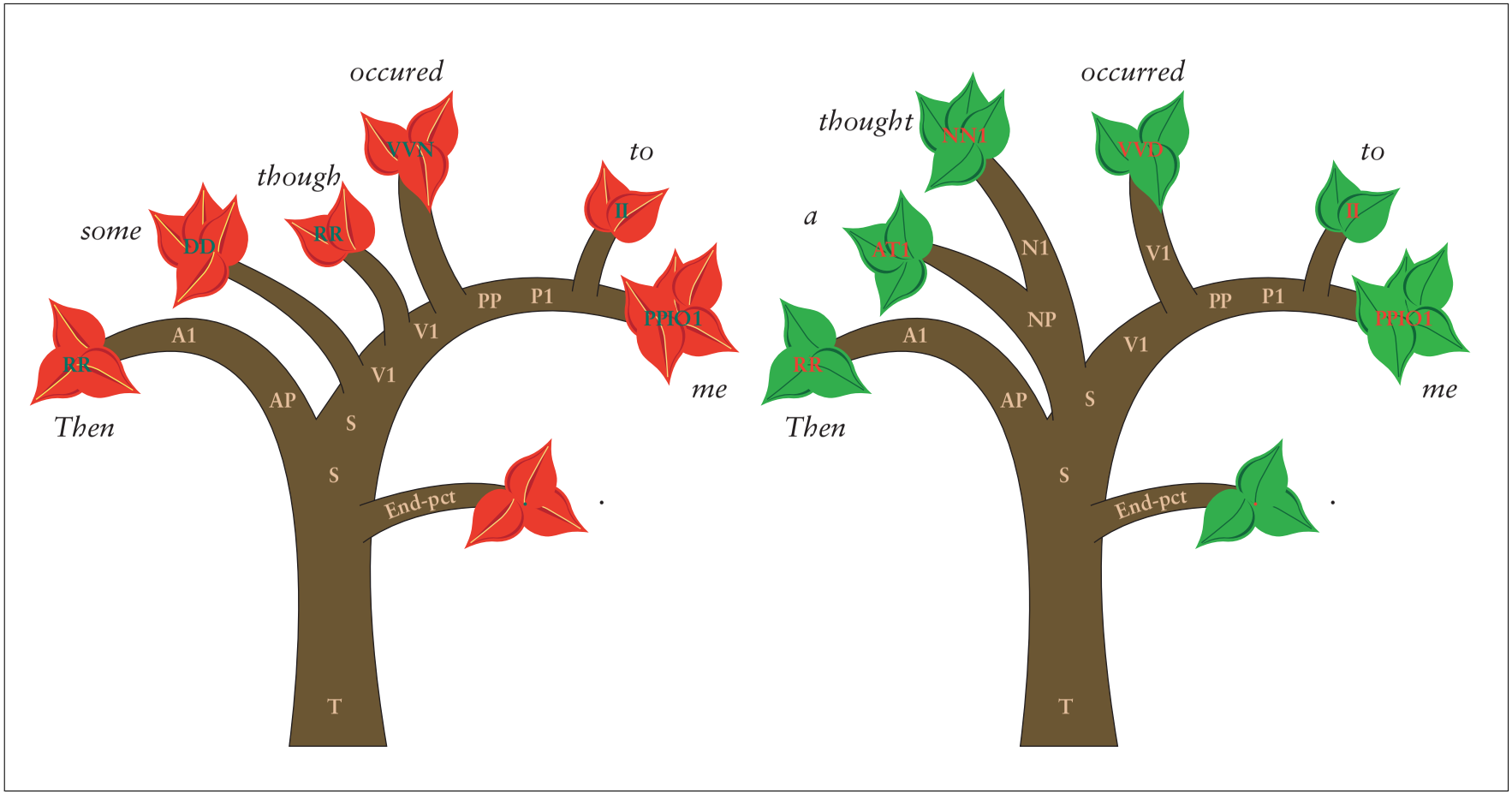

FIGURE 3.1. Parse trees grown by RASP from the sentences in EX. 10.

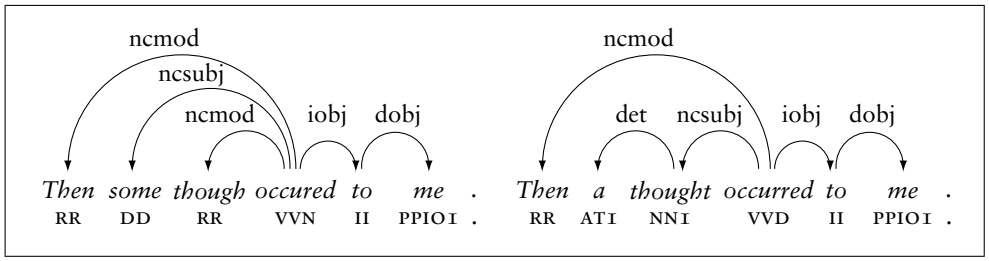

FIGURE 3.2. Grammatical relations derived from the parse trees for the sentences in EX. 10.

3.4. Parsing and metadata

As we have seen, the BNC contains word and sentence boundaries as well as part-of-speech tags (LEECH, GARSIDE & BRYANT 1994), but no parsing information and thus no facility to search for or otherwise make use of grammatical relations between words, which have proven useful in many applications. Various groups of people have parsed the corpus throughout the years using different tools and approaches. However, most, if not all, have simply removed all 'extraneous' mark-up from the corpus before parsing, which is not entirely satisfactory since we lose, *inter alia,* the distinction between titles and running text, formatting information, named entities and multi-word expressions, not to mention metadata including genre and provenance of texts and spoken data. (In addition, white-space modifications for tokenisation purposes will, if employed, cause further divergence from the original.) It seems to us that the only adequate solution is to keep the original mark-up intact and add new elements and attributes to indicate parsing information.

As an alternative to corpus-specific XML-handling code as had previously been used for the CLC, the option of integrating the different parts of the RASP system into an existing analysis framework able to handle XML was chosen, partly because this might reduce the amount of work required to parse a different corpus.

The Unstructured Information Management Architecture (UIMA) originated at IBM Research from a need to process initially unstructured data, mainly natural-language documents but also speech and images, with a series of complementary tools in order to produce structured data readily interpretable by a machine (FERRUCCI & LALLY 2004); for instance, the task of extracting a list of company names from a newspaper article might involve a named-entity recogniser which depends on the output from a part-of-speech tagger which in turn requires the original text to have been split into sentences and tokens. The lack of a standard for interoperability between different modules means that substantial effort is often required to make them work together; a well-defined architecture would solve this problem by allowing 'mixing and matching' of components without worrying about interfacing issues: each part adds new structured information in a predefined way, making it immediately available as input for the remainder of the processing chain. UIMA enables different tools to be encapsulated into modules which can, at least in principle, be combined

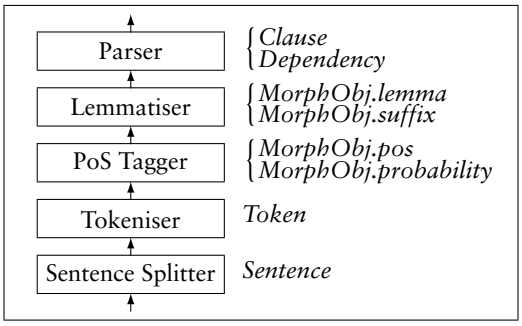

FIGURE 3.3. Analysis engines and annotations added.

freely and take advantage of existing modules for import and export of data. The Apache Software Foundation provides an open-source implementation[15].

3.5. RASP4UIMA

We[16] have made UIMA interfaces to the five individual components of the RASP system under the name RASP4UIMA, the first version of which is publicly available[17]. RASP's sentence splitter, tokeniser, part-of-speech tagger, lemmatiser and parser are hence available as separate *analysis engines* to all types of documents which can be handled within the UIMA framework. Fig. 3.3 provides a schematic overview of how each module contributes towards the final result.

Starting from unannotated text, the sentence splitter generates *Sentence* annotations; the tokeniser *Token*s; the part-of-speech tagger *MorphObj*s (potentially more than one for ambiguous tokens) with the features *pos* and *probability,* to which the lemmatiser adds *lemma* and *suffix;* and the parser *Dependency* relations and/or the full parse tree as *Clause* annotations. Each module uses annotations typically generated by the previous modules, but which may alternatively be obtained from elsewhere, like our using sentence boundaries and tokenisation information already present in the BNC.

We have also made an initial version of a *collection reader* and *consumer* (writer) to deal with the particulars of the BNC; the reader reads a document that adheres to the BNC-XML format, using the original mark-up to generate sentence and token annotations which can subsequently be used by the tagger, lemmatiser and parser (*see* Sect. 3.6 for BNC-specific details), whereas the consumer constructs a file containing all the original data as well as information obtained from processing.

As we shall see, multi-word expressions are currently parsed as individual words. RASP

[15]) http://uima.apache.org/

[16]) Original implementation by J. Nioche; BNC-specific and Unicode-related enhancements by me.

[17]) http://www.digitalpebble.com/rasp4uima/

handles many such expressions internally, but alternative approaches might assign such expressions a single part-of-speech tag or propose this as one alternative during parsing (*cf., e.g.,* LEWIN 2007 for experimental analysis of the utility of these different approaches). The RASP4UIMA framework is flexible enough to support these alternatives, though we have not implemented them in the first version.

RASP4UIMA actually contains a mechanism specifically designed to handle the case of mismatch between tokens and word units: part-of-speech tags are attached not directly to an atomic *token*, but instead to a higher-level *wordForm* which may correspond to any number of tokens. This representation of the linguistic information is inspired by the Morpho-syntactic Annotation Framework (MAF, *cf.* CLÉMENT & VILLEMONTE DE LA CLERGERIE 2005) which deals with morpho-syntactic annotation of specific segments of textual documents. MAF is currently under development as an ISO standard (ISO/DIS 24611).

3.6. Processing of the BNC

The BNC contains mark-up identifying sentences and tokens quite accurately, so it seems reasonable to take advantage of this information already present in the corpus rather than starting anew, which also alleviates the problem of how to combine pre-existing and additional annotations into an XML file at the end of the processing chain.

(11) <trunc>*Any*</trunc> *anyone who dissolved* <mw>*more than*</mw> ½
<gap desc="formula"/> *in rivers/lakes is n't gon na forget his pilgrimage , y'know .*

Ex. 11 shows an example sentence to which we are going to refer throughout this section. It has been artificially constructed from parts of actual sentences in the BNC to illustrate specific issues related to tokenisation, truncation, and so on. The full XML representation can be found in Appendix D.2; only the most essential mark-up is retained in Ex. 11, where tokens are indicated by separating white space instead of explicit mark-up.

3.6.1. Collection reader

Our BNC-specific collection reader uses a BNC-XML file as input to generate a UIMA representation of the textual content. Each sentence (*s* element) in the BNC results in a *Sentence* annotation, and words and punctuation marks (*w* and *c* elements, respectively) typically map to *Token*s. No exception is made for multi-word expressions, whose constituents are handled as ordinary tokens. We have occasionally found it necessary to depart from the tokenisation in the BNC, however, as not doing so would cause obvious problems:

First, a few thousand *w* and *c* elements are empty. These spurious elements were removed prior to parsing and are expected to be removed from a new official edition of the BNC as well.

Secondly, whereas most contracted forms like *isn't* and *cannot* have been split into two or several words as appropriate in the BNC, others like *let's* and *d'you* have not, nor have words separated by a slash like *his/her*. In such cases, what is marked up as one word in the BNC has nevertheless been treated as two or more tokens by the parser, as happens for the sequences *rivers/lakes* and *y'know* in the example.

Thirdly, in the BNC-XML, some parts of the original text or transcribed speech have been removed and replaced by a *gap* element, for reasons of anonymisation, inaudibility/illegibility, lack of appropriate textual representation, and so forth. In over 55 per cent of the cases, the redacted text is a name, address or telephone number, whereas tables and figures, illustrations and photographs, foreign material and somewhat complicated formulæ account for another 35 per cent. Such constituents typically play a syntactic role in the sentence and should not be ignored altogether; we have tagged all *gap*s as formulæ (&FO), which effectively means that they will be handled as noun phrases by the parser.

Finally, parsing spoken data presents specific challenges. The tagger lexicon has been extended to cover interjections and contractions not typically found in written text, but several speech-specific issues remain unaddressed. One particular problem is related to truncated words and false starts: sometimes, the speaker stops in the middle of a word, changes his mind and says something else, often as a replacement for the word he was about to utter as well as previous ones. Somewhat simplistically, we ignore truncated words (*i.e.*, words inside *trunc* elements). This does not always work out quite as nicely as in the example, but attempting to tag truncated words is unlikely to work well, so this seems like a reasonable approach given that the mark-up in the corpus does not encode the information required to reconstruct the complete/corrected utterance with false starts removed or relegated to parentheticals.

3.6.2. Part-of-speech tagging, lemmatisation and parsing

Once the UIMA representation has been generated, the tagger, lemmatiser and parser can be evoked normally, using the generic analysis engines. The current version of RASP, which includes enhancements originally developed for this project, is fully UTF-8 compliant, which means that tokens can be passed on directly without worrying about non-ASCII punctuation marks, accented letters, and so on. Example 12 shows the textual representation sent to the tagger.

(12) *anyone who dissolved more than ½ [gap] in rivers / lakes is n't gon na forget his pilgrimage , y' know .*

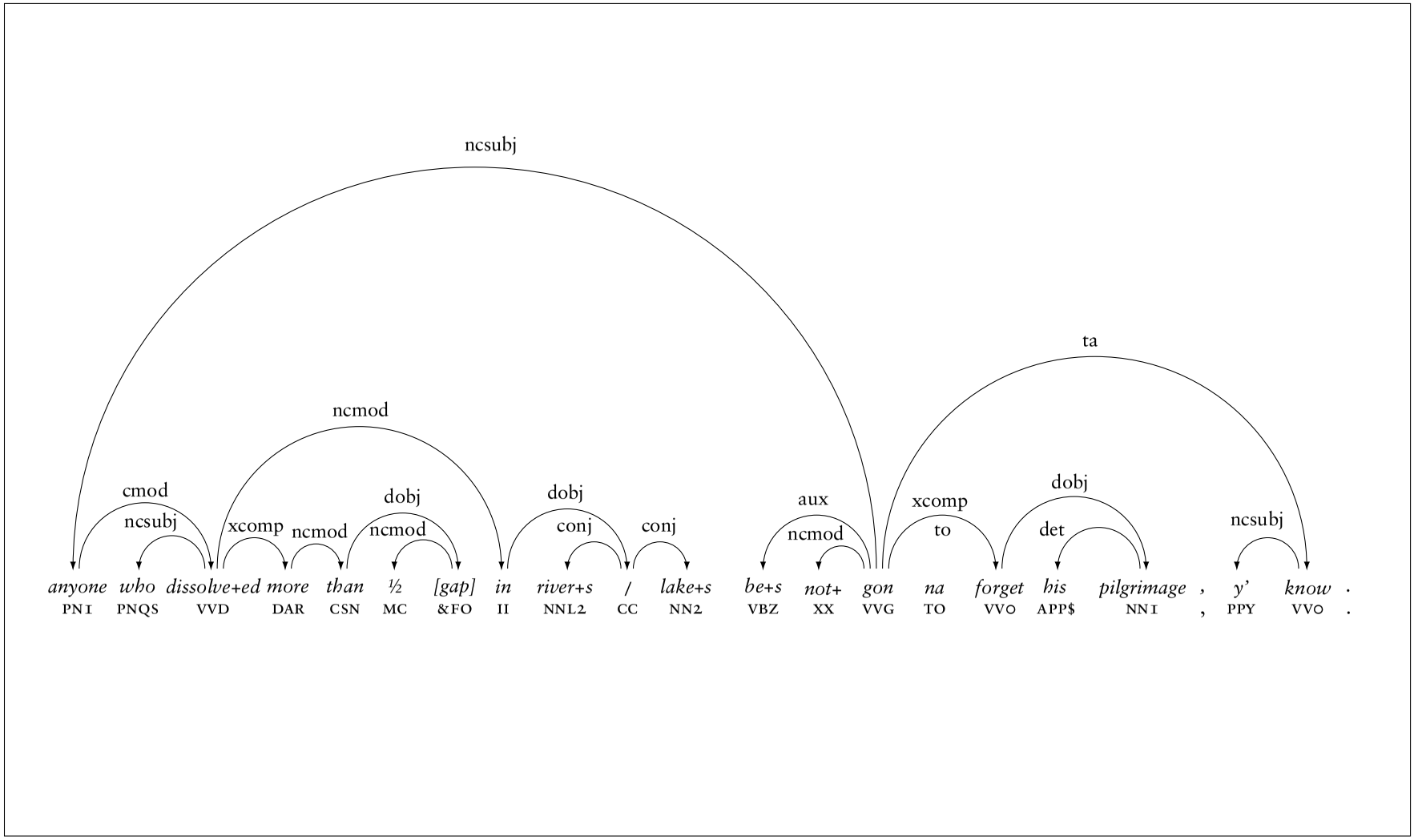

FIGURE 3.4. The tagger has added part-of-speech tags; the words have been lemmatised; and the parser has added dependency relations (GRs). The arrows representing GRs are drawn pointing from head to dependent; the label above the arrow indicates the type of relation, and the one below an optional subtype.

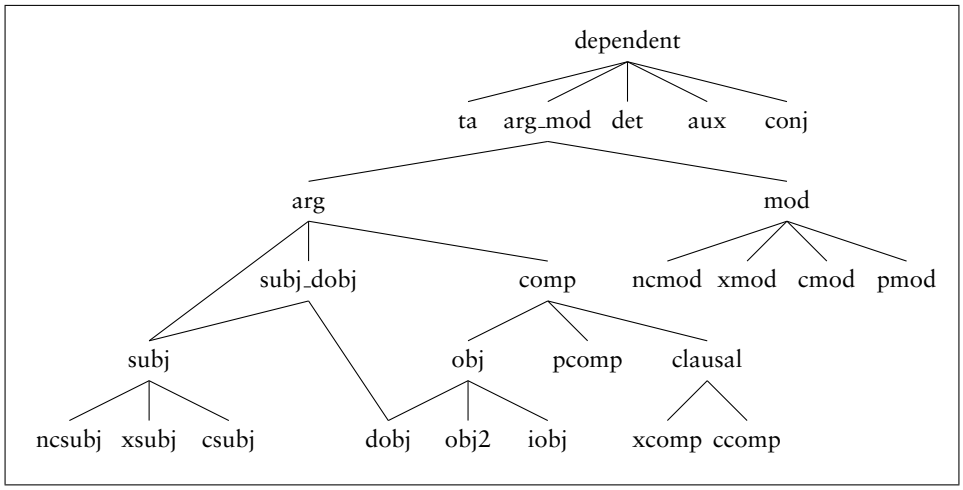

FIGURE 3.5. The GR hierarchy.

The information added to the sentence can be seen in Fig. 3.4. The parsing data is encoded not as trees, but as grammatical relations (GRs) between head and modifier. GRs result from transformation of a derivation tree constructed by the parser. The different relations are illustrated as a subsumption hierarchy in Fig. 3.5. They capture those aspects of predicate–argument structure that the system is able to recover and are the most stable and grammar-independent representation available. (*Cf.* BRISCOE 2006 for a more detailed description of the GR scheme.)

As there is no annotated test data for the BNC, we do not know how accurate the RASP analyses are. However, as we use an unlexicalised model, we expect performance to be similar to that on other out-of-training-domain test data (*c.* 80% GR precision and recall, *cf.* BRISCOE & CARROLL 2006 for details).

3.6.3. Collection consumer

A collection consumer written for the BNC uses the original BNC file as well as the newly generated annotations to create a new file containing the information from both sources as illustrated in Appendix D.2.2 which shows how the example sentence would end up.

The words have been numbered (attribute *id*) and annotated with part-of-speech tags (*rpos*) and lemma/suffix (*lem*/*affix*). The BNC already contains part-of-speech tags from a slightly less detailed tagset (*c5*) and a coarse word-class category (*pos*) as well as lemmatised forms derived using slightly different rules (*hw*); these cannot be used directly for parsing with RASP, but are kept in the corpus and may be useful for other applications, for instance to measure tagger agreement.

3.7. Distribution

We wanted to make the parsed version of the corpus available to others, which involved two major issues: we could not give the contents to anyone who did not already have a licence for the BNC itself, and it seemed wasteful to distribute the parsed corpus in its final form, including all the original data. Our initial plan was to rely on the Oxford Text Archive for distribution, since they would be in a position to know whether someone had a BNC licence, but it turned out to be more difficult than anticipated to work out the details and get an agreement in place. The obvious alternative was to make the parsed version available ourselves, but we did not feel comfortable with the idea that someone could gain access to the full BNC from us simply by pretending that they had a licence, and there was no easy way for us to check whether or not someone were a bona fide user of the BNC. We then came up with the idea of encrypting the parsed version using the original corpus as the encryption key; the data distributed by us would thus be useless to anyone not already in possession of the BNC itself. This method potentially puts the parsed version in the hands of someone who might have obtained the unparsed version by fraudulent means, but trying to prevent that seems pointless; our concern is simply that we should not facilitate unauthorised access to the original data.

The following sentence can be found in the BNC:

```
<s n="4">
   <w c5="AVQ" hw="how" pos="ADV">How </w>
   <w c5="VBZ" hw="be" pos="VERB">is </w>
   <w c5="NN1" hw="infection" pos="SUBST">infection </w>
   <w c5="VVN-VVD" hw="transmit" pos="VERB">transmitted</w>
   <c c5="PUN">?</c>
</s>
```

The parsed version includes the new information, but also everything found in the original:

```
<s n="4">
   <w n="1" c5="AVQ" hw="how" pos="ADV"
      rpos="NP1" lem="How">How </w>
   <w n="2" c5="VBZ" hw="be" pos="VERB"
      rpos="VBZ" lem="be" affix="+s">is </w>
   <w n="3" c5="NN1" hw="infection" pos="SUBST"
      rpos="NN1" lem="infection">infection </w>
   <w n="4" c5="VVN-VVD" hw="transmit" pos="VERB"
      rpos="VVN" lem="transmit" affix="+ed">transmitted</w>
   <c n="5" c5="PUN" rpos="?" lem="?">?</c>
   <grlist parse="1" score="-10.825">
         <gr type="ncsubj" head="2" dep="1"/>
         <gr type="xcomp" subtype="_" head="2" dep="3"/>
         <gr type="ncsubj" subtype="obj" head="4" dep="3"/>
```

```
        <gr type="xmod" subtype="_" head="3" dep="4"/>
        <gr type="passive" head="4"/>
  </grlist>
</s>
```

Since we want to make the parsed version available only to those who already have access to the unparsed version, there is no need to include the original data, but only the parts that have been added:

```
n="1" rpos="NP1" lem="How">How
n="2" rpos="VBZ" lem="be" affix="+s"
n="3" rpos="NN1" lem="infection"
n="4" rpos="VVN" lem="transmit" affix="+ed"
n="5" rpos="?" lem="?"
<grlist parse="1" score="-10.825">
      <gr type="ncsubj" head="2" dep="1"/>
      <gr type="xcomp" subtype="_" head="2" dep="3"/>
      <gr type="ncsubj" subtype="obj" head="4" dep="3"/>
      <gr type="xmod" subtype="_" head="3" dep="4"/>
      <gr type="passive" head="4"/>
</grlist>
```

The data can be made more compact by removing XML-style mark-up and instead using as separators characters which cannot occur verbatim in the data. We used line separator (\n), quotation mark (") and left angle bracket (<) as follows:

```
1"NP1"How
2"VBZ"be"+s
3"NN1"infection
4"VVN"transmit"+ed
5"?"?
1"-10.825<2"1"ncsubj<2"3"xcomp"_<4"3"ncsubj"obj<3"4"xmod"_<4""passive
```

As an additional space-saving measure, the format was constructed in such a way that optional attributes appear at the end and can often be left out completely when they do not appear in the corresponding XML (*e.g.*, the *affix* and *subtype* attributes in the example). An even more compact format could of course be devised, but we feel this is a good compromise between compactness and direct correspondence with the original format, which is crucial for straightforward reconstruction. The data was extracted and converted to this compact format using a Perl script and XML::Parser, whereas the encryption and, more importantly, the final decryption and merging which has to be effectuated by the end user, are implemented fairly efficiently in a small C program. The obvious alternative of compressing the encrypted files using a tool like *gzip* does not work well since the repetitive nature of XML gets obscured by the encryption, so the compact file format really contributes to considerably smaller files for distribution than could otherwise have been achieved.

It was expected that users of the corpus would want to decrypt and merge the entire corpus and save the complete files to disk. It turns out, however, that it may actually take longer to read a complete file from disk than to read the two smaller files and perform the decryption and merging: on our system, a time saving of 20–25 per cent was observed when *not* using pre-reconstructed files[18].

3.8. Limitations and further work

One of the major advantages of generic architectures such as UIMA is that they allow otherwise incompatible processing modules to be combined. In practice, inherent dependencies limit this somewhat; for instance, RASP's tagger can only be replaced by one that uses the exact same set of tags if one wants to use RASP's parser. Generic frameworks also involve a certain overhead: when we parsed the BNC, it became apparent that the resources available (most notably in terms of memory) did not permit us to parse with multiple tags assigned by the tagger within RASP4UIMA, despite our being able to do so when running RASP on a stand-alone basis. UIMA (and probably other similar sytems) also uses an internal representation which, despite being an application of XML, is not easily exploitable by other systems, so code has to be written to handle export of data to a different format (by means of a collection consumer) as well as import of data into the system in the first place (using a collection reader). Our expectation that a generic framework might almost automatically provide an interface to different corpora without a substantial amount of work for each individual corpus was thus not met.

For tasks where free 'mixing and matching' of different modules is not a goal, generic frameworks may be 'overkill', and the extra complexity and overhead they introduce may be better avoided, which in our particular case would be possible by enhancing RASP's native XML handling in such a way that it could read the BNC format as input and produce suitable output directly.

The annotation we have discussed so far has been added to the corpus, mixed with the original data ('in-line'),[19] a practice regarded by some as an unfortunate historic legacy that makes it difficult to verify that the original data remain unchanged and often complicated to retrieve the original, unannotated data from the corpus; the alternative is to leave the original data unadulterated and instead add the annotation to a separate file or as a separate layer ('stand-off', *cf.* THOMPSON & MCKELVIE 1997), which is now common practice (see PRZEPIÓRKOWSKI & BAŃSKI 2009 for an overview of of some of the major frameworks).

> Various representation standards and annotation tools have emerged over the past decade and have contributed to some convergence in practice, but at the same time, there has been growing

[18]) This result was obtained by outputting *c.* 1/10 of the files in the parsed BNC to */dev/null* or to the terminal.

[19]) The format developed for distribution of our parsed version of the BNC keeps the added data separate from the original, but only until the two are merged.

> recognition that *interoperability* among formats and tools, rather than universal use of a single representation format, is more suited to the needs of the community and language technology research in general.
>
> — IDE & SUDERMAN 2009

This has led to the development of 'pivot' formats such as the Graph Annotation Framework (GrAF; IDE & SUDERMAN 2007), intended to facilitate interoperability by providing a common representation.

Another advantage of stand-off annotation is that it automatically allows concurrent hierarchies (*e.g.,* the division of a book into into pages and columns on the one hand and into sentences and words on the other) even if the mark-up language imposes a tree structure (as does XML), simply by putting each hierarchy in a different file or on a different layer. This does however not solve the less common problem of truly non-hierarchical structures, a problem which can only be solved by using a more powerful mark-up language (DEROSE 2004; DI IORIO, PERONI & VITALI 2009).

To conclude this chapter, a few remarks on the XML format might not be out of place. A welcome uniformity at the mark-up level has been achieved thanks to SGML and XML (*see* Appendix B for comparison with pre-SGML formats), but there are still many options for the higher levels, and different annotation guidelines differ in their recommendations, so it is not the case in general that an XML corpus will be immediately comprehensible by a wide range of tools. For certain applications, a non-XML format may be appropriate, and the need for a widely understood format for data exchange should not unduly influence the choice of formats for internal representation; XML is to be regarded as an option amongst others, not as a panacea. We saw that our simple text-based compact distribution format not only resulted in smaller files, but also, somewhat unexpectedly, could be used to reconstruct the full XML files in less time (including decryption) than it would take to read them from disk. Another solution to XML's inherent inefficiency is to use an alternative serialisation devised for increased processing efficiency and compactness, such as the Efficient XML Interchange (EXI) Format (SCHNEIDER & KAMIYA 2009), which can be seen as introducing the efficiency of a task-specific binary format into the XML world.

Chapter 4.

Binary sentence classification

As we saw in chapter 2, commercial grammar checkers typically rely on hand-crafted rules to detect a restricted set of errors. Atwell described an early attempt to avoid this (1987), and others have trained classifiers on artificial errors. Foster argued that genuine samples are needed (2004), but the idea of training a classifier using real-life examples of incorrect constructions was merely suggested, as it would require a much larger corpus than the one she had compiled.

The clc provides a large quantity of English text with a high error rate, and the error annotation identifies the errors and gives suitable corrections, which makes a supervised machine learning approach feasible.

Reliable identification, classification and correction of each individual error in a text is a difficult task; not even a human expert is able to provide reliable corrections for errors which obscure the author's intended meaning, and even determining whether a piece of text is correct or not may require detailed knowledge of the subject matter. Verification of individual words out of context, on the other hand, would amount to spelling rather than grammar checking. Classification of individual sentences as being either correct or incorrect is situated somewhere between the two in difficulty and can be approached with machine learning techniques, using features extracted from correct and incorrect sentences as training data for supervised learning.

4.1. Machine learning

In general terms, a supervised classifier is trained using a set of couples $(\mathbf{f}, c)$ consisting of a feature vector $\mathbf{f}$ of features f_i associated with a class c; subsequently, the classifier

should be able to assign an unseen vector $\mathbf{f}'$ to the class c' to which it belongs. The couples $(\mathbf{f}, c)$ can be said to constitute the experience from which the machine has been able to 'learn' appropriate generalisations, whence the term 'machine learning'. For the binary sentence classification task, there are two classes, correct and incorrect sentences, and each feature vector is intended to give a salient representation of the corresponding sentence, typically encoding characteristics such as the individual words from which it is made up, combinations of adjacent or grammatically related words, parts of speech and sentence length.

The selected machine learning algorithms are the ones previously shown to perform well on this task (ANDERSEN 2006). Compared to previous work, the results reported in the following are obtained using cleaner data (in particular, excluding erroneous sentences for which no correction is given, which were previously treated as effectively being both correct and incorrect), and all the classifiers have been trained on the same amount of data. The Machine Learning for Language Toolkit (MALLET)[20] provided implementations of all the machine learning algorithms used, apart from the support vector machine which was instead provided by the SVM-perf[21] implementation (JOACHIMS 2006).

4.1.1. Naïve Bayes

According to Bayes' rule, the probability that a sentence described by a feature vector $\mathbf{f}$ belong to a given class c, $P(c|\mathbf{f})$, can be reformulated as follows:

$$P(c|\mathbf{f}) = \frac{P(\mathbf{f}|c)\,P(c)}{P(\mathbf{f})}$$

$P(\mathbf{f})$ is constant for a given sentence and need not be evaluated if we are only interested in finding the most likely c,

$$\hat{c} = \operatorname*{argmax}_c \frac{P(\mathbf{f}|c)\,P(c)}{P(\mathbf{f})} = \operatorname*{argmax}_c P(\mathbf{f}|c)\,P(c).$$

The conditional probability $P(\mathbf{f}|c)$ is unfortunately difficult to estimate directly from training data, due to data sparsity (unique $\mathbf{f}$ for most distinct sentences unless the feature set is extremely restricted); the standard solution is to assume that all the features f_i in $\mathbf{f}$ are independent, which allows us to express the most likely c as

$$\hat{c} = \operatorname*{argmax}_c P(\mathbf{f}|c)\,P(c) = \operatorname*{argmax}_c P(c) \prod_i P(f_i|c).$$

It would of course be the pinnacle of naïveté to believe that the words in a meaningful sentence are completely unrelated, but the strong independence assumption underlying naïve Bayesian classification does not actually hamper performance at all to the extent one might have expected.

[20]) http://mallet.cs.umass.edu/

[21]) http://svmlight.joachims.org/svm_perf.html

4.1.2. Balanced winnow

LITTLESTONE introduced a new machine learning algorithm (1988), called winnow by analogy with the agricultural method used for separating grain from chaff because it is 'designed for efficiency in separating relevant from irrelevant attributes' and thus learn from data with a portentially large proportion of irrelevant attributes, a task that is particularly important in high-dimensional feature spaces such as the ones that appear when words and n-grams of words are used as features.

The balanced winnow algorithm is parameterised by an update parameter ε and a near-miss threshold δ and maintains, for each class c, a weight vector $\mathbf{w}_c$ with the same number of dimensions as the number of features and initialised to $(1, 1, \ldots, 1)$. The classifier predicts the class

$$\hat{c} = \operatorname*{argmax}_c \mathbf{w}_c \cdot \mathbf{f} = \operatorname*{argmax}_c \sum_i w_{c,i} f_i.$$

During training, if the predicted class $\hat{c}$ is equal to the actual class c, the classifier has not made an error, and nothing needs to be done; if, on the other hand, the prediction is incorrect, $\hat{c} = c' \neq c$ (or, at least in the MALLET implementation, if the classifier almost chose the second best class $c' \neq c$ instead, as defined by the inequality $\mathbf{w}_c \cdot \mathbf{f} - \mathbf{w}_{c'} \cdot \mathbf{f} = (\mathbf{w}_c - \mathbf{w}_{c'}) \cdot \mathbf{f} < \delta$), the weight vectors $\mathbf{w}_c$ and $\mathbf{w}_{c'}$ need to be updated in order to reflect the training data more accurately at the next step $k + 1$:

$$\forall i, \qquad w_{c,i}^{(k+1)} = (1 + \varepsilon)^{f_i} w_{c,i}^{(k)} \qquad \text{and} \qquad w_{c',i}^{(k+1)} = (1 - \varepsilon)^{f_i} w_{c',i}^{(k)}.$$

(Given $(1 \pm \varepsilon)^0 = 1$, only the weights corresponding to non-zero features $f_i \neq 0$ in the misclassified instance will be updated.) The original formulation used $(1 + \varepsilon)^{-f_i}$ instead of $(1 - \varepsilon)^{f_i}$, which makes the update procedure symmetrical. The otherwise similar Perceptron algorithm differs by doing additive updates instead of multiplicative ones.

4.1.3. Maximum entropy

The principle of maximum entropy, as formulated by JAYNES, says that

> in making inferences on the basis of partial information we must use that probability distribution which has maximum entropy subject to whatever is known. This is the only unbiased assignment we can make; to use any other would amount to arbitrary assumption of information which by hypothesis we do not have.
>
> — 1957

A classifier based on this principle is trained in such a way as to maximize the entropy

$$H(p) = -\sum_{\mathbf{f},c} p(\mathbf{f}, c) \log p(\mathbf{f}, c)$$

whilst remaining consistent with the partial evidence available in the form of training data. In the case of no training data available, this will give a uniform distribution. Given feature vectors with binary features f_i, the constraints imposed by the training data can be expressed by the set of equations

$$\forall i, \quad E_p[f_i] = E_{\tilde{p}}[f_i],$$

where

$$E_p[f_i] = \sum_{\mathbf{f},c} p(\mathbf{f}, c)\, f_i$$

is the expectation of f_i predicted by the model, and

$$E_{\tilde{p}}[f_i] = \sum_{\mathbf{f},c} \tilde{p}(\mathbf{f}, c)\, f_i$$

is the empirical expectation of f_i evaluated on the training data. According to the principle of maximum entropy, the best model amongst the ones that satisfy the constraints,

$$P = \{p \mid \forall i,\ E_p[f_i] = E_{\tilde{p}}[f_i]\},$$

is the one of maximum entropy, namely

$$p^* = \operatorname*{argmax}_{p \in P} H(p).$$

4.1.4. Support vector machines

Support vector machines (SVMs) are based on the idea of a hyperplane H separating the data into two classes (assuming that the classes are actually linearly separable). A hyperplane can in general be identified by a normalised perpendicular vector n and the distance d from the origin; the hyperplane contains any point x satisfying

$$\mathbf{n} \cdot \mathbf{x} - d = 0.$$

Given a set of training vectors $(\mathbf{f}_i, c_i)$ with $c_i \in \{-1, +1\}$, the condition that the hyperplane must separate the classes can be expressed as

$$\forall i,\ c_i(\mathbf{n} \cdot \mathbf{f}_i - d) > 0.$$

In order to maximise the chances that unseen instances will fall on the appropriate side of the hyperplane, we want to maximise the distance between the training vectors and the hyperplane. To achieve this, we may introduce two auxiliary hyperplanes H^+ and H^-, parallel to H and shifted by $\pm\delta$, and require that they satisfy the same condition as H, which gives the expression

$$\forall i, \begin{cases} c_i(\mathbf{n} \cdot \mathbf{f}_i - (d + \delta)) > 0; \\ c_i(\mathbf{n} \cdot \mathbf{f}_i - (d - \delta)) > 0 \end{cases}$$

or, equivalently,

$$\forall i,\ c_i(\mathbf{n} \cdot \mathbf{f}_i - d) > \delta.$$

The sought hyperplane is obtained by maximising δ, and the vectors that limit δ and thus prevent H^+ and H^- from moving further apart are called support vectors.

An important generalisation is the substitution of the scalar product by a kernel function, which essentially transforms the problem into a higher-dimensional space in which the vectors will be linearly separable if an appropriate kernel function is chosen. Another extension is the use of a soft margin, which in particular makes the method more robust with respect to misclassified training instances which would otherwise prevent a separating hyperplane from being found, formalised by introducing slack factors ξ_i, which gives

$$\forall i,\ c_i(\mathbf{n} \cdot \mathbf{f}_i - d) > \delta - \xi_i.$$

We still want to maximise δ, but at the same time minimise the accumulated degree of misclassification $\sum_i \xi_i$.

CORTES & VAPNIK provide a detailed description of SVM techniques (1995).

4.2. Sentence selection and feature extraction

Data for training and testing was obtained from the parsed version of the CLC (*cf.* Sect. 3.3). In order for the classifier to be able to learn significant differences between correct and incorrect sentences, as opposed to coincidental correlation between vocabulary and correctness that might exist in the training data simply because certain words happened not to occur in both correct and incorrect sentences, only sentences for which both a correct and an incorrect version exist were selected. The resulting training and test sets were therefore balanced in the sense that they contained the same number of correct and incorrect sentences. The selection procedure excluded both sentences without errors and those for which no correction was provided, as well as sentences that had been split or merged, typically because a comma had been changed into a full stop or *vice versa*.

Fig. 4.1 gives a summary of the features found to be useful for this sentence classification task: morphologically analysed words and part-of-speech tags used in the *n*-gram features are taken from the parse tree, combined with adjacent ones as necessary to form bigrams and trigrams; unlabelled grammatical relations are simply extracted from the parser output; the binary feature indicating a parsing error is set if there is no complete parse for the sentence; and the fragment features indicate whether the parse was created using a fragment rule and, in that case, the type of its constitutive fragments.

The possibilites for extraction of features from the parsed sentences are by no means exhausted; some obvious extensions include features combining words and part-of-speech

Feature	Examples
Word	*a; thought; occur+ed*
Word bigram	*a thought; thought occur+ed*
Part of speech	ATI; NNI; VVD
Part-of-speech bigram	ATI NNI; NNI VVD
Part-of-speech trigram	ATI NNI VVD
Grammatic relation (words)	*thought→a; occur+ed→thought*
Grammatic relation (parts of speech)	NNI→ATI; VVD→NNI
Parsing error (binary)	parsing error
Sentence fragments (binary)	fragment
Sentence fragments (clause type)	NP fragment; PP fragment

FIGURE 4.1. Overview of the features used. The examples in the first part of the table are taken from the fragment a_{ATI} $thought_{\text{NNI}}$ $occurred_{\text{VVD}}$.

Classifier	Training data	Test data
Naïve Bayes	80.36%	70.39%
Balanced winnow	95.25%	69.76%
Maximum entropy	79.41%	71.27%
Support vector machine	73.65%	70.88%

FIGURE 4.2. Accuracy (*i.e.,* the proportion of sentences correctly identified) for different classifiers, all trained on 346,112 sentence pairs, each comprising a correct and an incorrect version of the same sentence, and tested on 43,352 pairs. All differences in test accuracy are statistically significant, thanks to the large number of test sentences. The size of the SVM model was restricted by available memory, which is likely to have limited its performance. This issue was not investigated in any detail, however, since thorough comparison of different machine learning techniques was not the primary focus of the research.

tags to form 'mixed' *n*-grams, as well as ones encoding the type of grammatical relation existing between a pair of words or part-of-speech tags.

4.3. Classification performance and analysis

Several classifiers were trained and tested using the data described in the previous section. Fig. 4.2 shows the accuracy obtained for different machine learning algorithms on the task of classifying sentences as being either correct or incorrect. There is a lot of variation in performance on the training set, but the classifier that does best on the training set is the one that does worst on the test set, so it seems to have achieved this by learning idiosyncrasies rather than general characteristics. All the classifiers achieve an accuracy of around 70 per cent on the test set; the maximum entropy classifier gets the best result with 71.27 per cent accuracy, although it is possible that the SVM classifier could have done as well or better if it were not for memory limitations.

The accuracy is highly dependent upon the type of error, as indicated in Fig. 4.3. Spelling mistakes obtain the highest detection rate, closely followed by inflectional and derivational errors, all of which tend to result in intrinsically malformed words. Verbal tense errors are

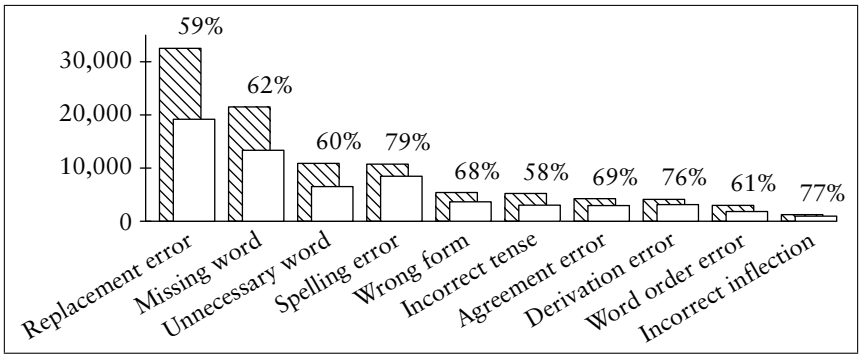

FIGURE 4.3. The hatched bars represent the total number of errors of each category in the test set, and the open bars represent the number of errors occurring in sentences identified as incorrect by the classifier. The percentage shows the proportion of errors that occur within sentences classified as erroneous. It might be worth pointing out that the classifier cannot in general be assumed to have identified each individual error explicitly: in a sentence with multiple errors, each of them may result in characteristic features which only together enable the classifier to detect that the sentence is incorrect; moreover, an error that does not engender any discernible features will nevertheless seem to have been detected if the sentence in which it appears has been identified as incorrect as the result of another error. (Data from the naïve Bayes classifier.)

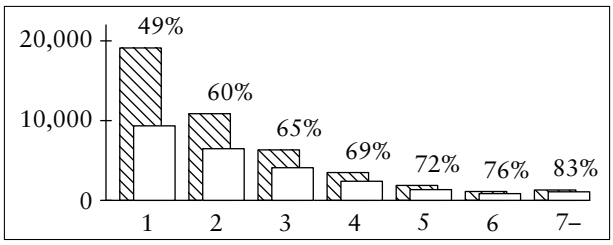

FIGURE 4.4. The hatched bars indicate the total number of sentences with a given number of errors in our test set, and the open bars indicate the ones that are correctly classified as erroneous. The percentage is the proportion of incorrect sentences correctly identified as such. (Data from the naïve Bayes classifier.)

located at the opposite end of the spectrum, and the system hardly does better on errors involving replacement, insertion or deletion of words; these errors are clearly harder to spot, not only because the mistake consists in a prohibited combination of individually correct words, but also because the wrong sentence may be grammatically correct, merely conveying a message different from the one the writer intended.

Accuracy as a function of the total number of errors per sentence is shown in Fig. 4.4, which clearly indicates that a higher number of errors in a sentence makes it more likely to be caught by the system. This is hardly surprising, given that a sentence with multiple errors will contain many discernible features, whereas a sentence with one single, possibly minor, mistake will have much in common with a perfectly correct sentence. Conversely, a correct sentence does not differ substantially from a slightly incorrect one and may thus mistakenly be classified as incorrect.

The strong effect on performance caused by the mere quantity of errors in a given sentence suggests that we may want to look at potential errors individually, which would also make

Feature	Examples
Article + letter	*a g; an u; a u; an i*
Article + word	*a green; an unusual; a useful; an iron*

FIGURE 4.5. Overview of the features used. The examples are taken from the phrases *a green man, an unusual task, a useful tool,* and *an iron,* all of which are correct.

it possible to use more directed approaches to various kinds of errors. An entire sentence would then be classified as incorrect if at least one error was identified with a high degree of confidence.

4.4. Quantitative effects

In the sentence-classification experiment described, there are many sources of noise and several things going on at the same time, which makes it difficult to identify the different contributions and draw clear conclusions. We therefore wanted to explore the influence of the number of errors per sentence on classification performance in a more controlled setting in order to check whether the quantitative effect remains when the naïve Bayesian classifier is given as input a much more restricted set of features aimed at detecting a single type of error.

For this experiment, we chose to look at the relatively simple problem concerning the form of the indefinite article, which shall be either *a* or *an* depending on whether the following sound is consonantal or vocalic. There is a certain discrepancy between spelling and pronunciation (consider, *e.g.*, *an hour*, *an MP* and *a European*), so the correct form is not immediately obvious from the following letter. We therefore extracted two distinct features for each occurrence of the indefinite article: 1) the article combined with the following letter, and 2) the article combined with the following word (*see* Fig. 4.5). Thanks to the CLC, both correct and incorrect examples could be extracted and used for training.

The initial result on our test set is indicated in Fig. 4.6 and corresponds to 55 per cent recall of incorrect sentences with 80 per cent precision, which is hardly impressive on such a simple task. When we looked more closely at the data, however, it turned out that many of the overlooked errors occurred within sentences with multiple instances of the indefinite article; in fact, 71 per cent of the incorrect sentences that contain one or more additional, correct, instances of the indefinite article are incorrectly classified as correct, as are 82 per cent of the sentences with two or more such instances. A possible explanation could then be that the evidence for an error was simply drowning in evidence for the contrary, making the sentence 'predominantly correct' in the probabilistic eyes of the classifier.

We then ran a new experiment, in which not entire sentences, but single occurrences of the indefinite article were to be classified as correct or incorrect. As indicated in Fig. 4.7, 95 per cent of the incorrect occurrences were found, and very few correct occurrences were

Classification	Correct	Incorrect
Correct ($p > 0.5$)	162,394	166
Incorrect ($p > 0.5$)	51	199

FIGURE 4.6. Sentence-level classification. Results shown as number of sentences. The correct sentences include those that do not feature the indefinite article, but they are all classified as correct by the classifier, so this does not affect precision and recall of incorrect sentences.

Classification	Correct	Incorrect
Correct ($p > 0.5$)	26,947	17
Incorrect ($p > 0.5$)	34	314

FIGURE 4.7. Classification of each occurrence of the indefinite article.

mislabelled (90% precision). This seems to indicate that individual examination of potential errors may indeed be fruitful.

A closer inspection of the classifier output shows that the classifier correctly classified the vast majority of the data with high confidence ($p \geqslant 0.9$ for the chosen class label), as indicated in Fig. 4.8; it missed only two errors (**a HTV* and **a MC*) and apparently gave ten false positives with this high degree of confidence. However, four of the 'false' positives turn out to be real errors overlooked by the annotators, one is due to a transcription error (*can* transcribed as *c an* with a superfluous space), and the remaining are occurrences of the letter *a* rather than the indefinite article *a*, a problem that could have been avoided using part-of-speech tagging. The quasi-totality of the misclassifications are thus done with a lesser degree of confidence and mainly concern somewhat irregular or difficult cases like *underground*, *universal*, *US* and *historic*; these can be checked manually, or the classifier can be improved, be it by means of more training data or by using more salient features.

4.5. Expert error detectors

After having demonstrated that concentrating on single errors could indeed be beneficial, we wanted to attack a slightly more complex error type, requiring some knowledge of grammar as opposed to mere juxtaposition of words. In this section, we look at a deterministic system for error detection, with no machine learning involved. Agreement between determiner and noun was chosen as a well-defined, purely grammatical, sentence-internal problem

Classification	Correct	Incorrect
Correct ($p \geqslant 0.9$)	26,872	2
Correct ($p < 0.9$)	75	15
Incorrect ($p < 0.9$)	24	22
Incorrect ($p \geqslant 0.9$)	10	292

FIGURE 4.8. The results from Fig. 4.7 broken down by the probability for the chosen class as calculated by the classifier.

Relation involving *this*	Correct	Incorrect
No link to plural noun	48,590	326
Link to plural noun	326	1,105

FIGURE 4.9. Presence or absence of a link between *this* and a plural noun in correct and incorrect instances of the singular determiner.

that does not rely on the meaning of the words involved or other less clear-cut concepts, and we then focused on the misuse of *this* when *these* is needed, by far the most common confusion of this kind in the CLC.

(13) a. **this/these friends*

b. **this/these old school friends*

(14) a. *This is what good friends do.*

b. *I need this/these for a meeting with good friends of hers.*

Ex. 13 shows that we really need to identify the noun determined by *this/these* and examine its number in order to be able to choose between the singular and the plural determiner. Moreover, *this/these* does not always determine a noun at all, as illustrated by Ex. 14, and in this case no determiner-noun agreement should be attempted.

The RASP parser is able to identify the grammatical relation between *friends* and *these* in both phrases in Ex. 13, and it correctly refrains from establishing a direct link between *this/these* and any of the nouns in Ex. 14. Because the RASP system is aware of agreement rules, no connection will be made between the plural noun *friends* and the incorrect singular determiner *this* in Ex. 13, but this is, alas, not a reliable indication of error,[22] given that *this/these* does not always determine a noun. A possible solution is to make RASP ignore the requirement of number agreement between noun and determiner; then, the system will allow a singular determiner like *this* to act on a plural noun like *friends*, and the presence of a grammatical relation between a singular determiner and a plural noun (or *vice versa*) should be a good error indication.

We parsed the sentences containing *this* with this slight modification to the parser and found that the impossible relation between singular determiner and plural noun was established in 77 per cent of the sentences containing an incorrect instance of *this*, and in only a very low proportion of the full set of correct sentences (*see* Fig. 4.9). Quite a few of the incorrect instances were actually impossible to spot due to interactions with other errors

[22]) If, on the other hand, no grammatical relation contains *this*, then something is probably amiss. Only a small portion (at most 20% in our experiments) of the incorrect sentences can be found by exploiting this directly, though, for the parser will often find another (incorrect) grammatical relation involving *this*.

in the text, as for instance *this job* corrected to *these jobs*; we therefore re-evaluated the system's performance on the set of incorrect sentences in which the determiner agreement error appears in isolation, which gave a recall of 92 per cent or 197 out of 215 incorrect uses of *this* correctly identified as such. The remaining 8 per cent were missed partly due to parsing or tagging errors (*e.g.*, an instance of the plural noun *treasures* was tagged as a third person singular verb form, which in turn prevented the correct analysis from being found), and partly due to genuine number ambiguity (*this/these people*).

4.6. Direct comparison of classification results

As we have seen, the initial system performs relatively poorly on sentences containing a single error (*see* Sect. 4.3), whereas specialised classifiers for specific error types can perform well. In this section, we present a more direct comparison between the two approaches by evaluating performance on sentences containing exactly one error of a type for which a specialised classifier has been developed. In addition to *a/an* and *this/these* confusions, spelling errors were added to the set, partly to compare the naïve machine learning approach with simplistic dictionary look-up, partly to be able to evaluate the other classifiers' false-positive rate.

General performance for the initial naïve Bayes classifier is reported in Fig. 4.2–4.4; more specific data in the form of confusion matrices for the three types of error under consideration here can be found in Fig. 4.10.

4.6.1. Individual classifiers independently

Fig. 4.11 shows the results obtained using a specialised classifier for each individual error type. This particular evaluation method relies upon the assumption that the potential type of error is known *a priori,* which is clearly not true in general. We have therefore chosen not to quote overall performance, since this would have been purely artificial.

As expected, determiner-form errors were detected reliably: only one occurrence of **a honest* slipped through, and an instance of **a English* which had not been marked up by the annotators was correctly flagged as incorrect.

The results for determiner-agreement errors appear less conclusive, but the somewhat low recall has an explanation: whilst lack of agreement between a noun and *this/these* is the predominant error, the category also comprises noun-agreement errors with *that/those*, as well as other agreement errors, including those involving pronoun–antecedent agreement in gender *(his/her)* and number *(its/their)*.

Spelling mistakes were identified using a list of words derived from a small Oxford dictionary (Mitton 1986) to which have been added 50 frequent words, mostly neologisms like

	Classification	
	Correct	Incorrect
Determiner-form error		
Corrected	44	13
One error	35	22
Determiner-agreement error		
Corrected	35	3
One error	10	28
Spelling mistake		
Corrected	1,421	374
One error	577	1,218
Corrected (Σ)	1,500	390
One error (Σ)	622	1,268

FIGURE 4.10. Confusion matrices for the initial system (the naïve Bayes classifier for which overall accuracy is reported in Fig. 4.2). The columns 'correct' and 'incorrect' indicate, for each set of corrected/incorrect sentences, the number of sentences that were classified as being either correct or incorrect. Each confusion matrix shows how well the classifier is able to distinguish between sentences containing one error on the one hand and their corrected counterparts on the other hand. The bottom matrix is the sum of the three others, indicating the total performance on these three error types, which can also be expressed as 76% precision and 67% recall.

	Classification	
	Correct	Incorrect
Determiner-form classifier		
Corrected	55	0
One determiner-form error	1	54
Determiner-agreement classifier		
Corrected	38	0
One determiner-agreement error	19	19
Spelling-mistake classifier		
Corrected	1,747	48
One spelling mistake	381	1,414

FIGURE 4.11. Confusion matrices for the individual classifiers. The set of sentences is the same as in Fig. 4.10, but instead of using a general classifier, a specialised one is used for each type of error. Each classifier only deals with sentences containing the type of error that it has been designed to detect, as well as corrected versions of those (*i.e.*, the classifiers do not yet have to handle sentences containing other types of error.)

website and *fax* or alternative spellings like *cafe* without an accent and *organisation* with an *s*. Lower-case words not in the list are considered to be incorrect, whereas words containing upper-case letters are not checked at all; consequently, a few correct words (mostly somewhat rare words or non-Oxford variants) were flagged as incorrect, whereas a more important number of words written with an initial capital (only a minority of which are proper nouns) or entirely in upper case went unnoticed. This highly unoptimised system actually performed better than the general classifier, which is probably due to the fact that the latter can only detect misspelt words that occur in the training data.

	Classification	
	Correct	Incorrect
Determiner-form classifier		
Corrected	1,887	3
One error	1,835	55
Determiner-agreement classifier		
Corrected	1,886	4
One error	1,865	25
Spelling-mistake classifier		
Corrected	1,840	50
One error	474	1,416
Corrected (Σ)	1,833	57
One error (Σ)	401	1,489

FIGURE 4.12. Confusion matrices for the individual classifiers on the full test set. As opposed to in Fig. 4.11, each classifier is given the full set of sentences for consideration and is supposed to pick out the sentences that contain the type of error on which it specialises. Here, 'correct' only means that the classifier in question did not find an error, not necessarily that the entire sentence should be regarded as impeccable, as there could well be other types of error detected by other classifiers. A few sentences are flagged as incorrect by more than one classifier, which in particular explains that the total number of incorrect sentences correctly identified as such is lower than the sum of the corresponding numbers provided by each of the individual classifiers. In summary, the system gives 79% recall with 96% precision on the test set.

4.6.2. Individual classifiers together

What happens when the individual classifiers are let loose on the entire test set? A higher number of false positives can be expected, but as shown in Fig. 4.12, this effect is relatively modest. Sentences within which an error is found by at least one of the classifiers are considered to be incorrect; the remainder are taken to be correct. Combining the evidence in this way gives a system with an *F*-score of 87 per cent on the test set, as compared to 71 per cent for the initial system.

4.6.3. Specialised classifiers providing additional features

Nothing prevents us from using discriminative information obtained from the individual classifiers as features for the initial machine-learning system. We tried to add a feature indicating whether or not a spelling mistake was detected in order to see whether additional information can actually be beneficial without any other changes to the system. (Features corresponding to the other classifiers could of course have been added as well, but given the relative small number of errors affected, doing so would not have had much influence on the overall performance.)

As can be seen from Fig. 4.13, the performance in terms of accuracy on the test set for the three machine learning algorithms we tried increased with 1–1.5 percentage points when this new feature was added. For the naïve Bayes classifier, the improvement corresponds to

Classifier	Training data	Test data
Naïve Bayes	80.47%	71.31%
Balanced winnow	95.33%	71.39%
Maximum entropy	77.80%	72.41%

FIGURE 4.13. Accuracy on an experiment similar to the one from Fig. 4.2, the only difference being the addition of a new binary feature indicating likely misspellings. Each classifier performs statistically significantly better when this feature is added. The difference in accuracy between the naïve Bayesian and the balanced winnow classifiers is not statistically significant, whereas the one between either of them and the maximum entropy classifier is.

an additional 391 sentences correctly classified, as compared to a reduction of misclassified sentences by 518 when the specialised classifier was used directly. Given the amount of noise and the number of features used by the classifier, this effect of adding one additional feature seems rather encouraging.

4.7. Conclusion

An error-annotated corpus provides sentences known to be incorrect; when corrections are provided, like in the CLC, correct and incorrect versions of the same sentence can be extracted, which makes it easier for a machine learning algorithm to identify salient characteristics of correct and incorrect sentences, respectively. As a useful addition to individual words and *n*-grams, a parser provides more general features (parts of speech) as well as more linguistically motivated ones (grammatical relations).

We have seen that a classifier trained on correct and incorrect sentence pairs can give a binary sentence classification accuracy of over 70 per cent. As one might expect, the system is more successful at handling morphological and typographical errors than syntactic ones, and the lowest success rate is observed for verb tense errors, which are likely to depend on extra-sentential context. The extent to which errors can be detected reliably in individual sentences will be discussed further in Chapter 7, as will the problem of inconsistent annotation in the corpus.

Our analysis of the classification performance also showed that the system was able to detect highly deviant sentences with high accuracy, but was less successful with sentences containing but a single error. It would be interesting to see whether excluding sentences with more than one error or with errors spanning several words from the training data might be beneficial, although this might exclude certain types of errors which seldom appear on their own. Another solution, and the one we have presented as successful, consists in detecting specific types of error directly. Such expert detectors may be used directly or used as an additional source of information for sentence classification.

Chapter 5.

Synthetic errors

We saw in the previous chapter that pairs of correct and incorrect versions of sentences extracted from a corpus can be used for training and evaluation of a system that distinguishes between correct and incorrect sentences. A learner corpus like the CLC is a good source of errors for a classifier aimed at detecting human errors, in particular the ones committed by foreign learners of English, since it contains real errors produced by this demographic group. An alternative is to use ungrammatical data generated automatically from a corpus of correct sentences, more readily available and generally larger than an annotated learner corpus. However, such artificial data are of little use if the errors are not sufficiently similar to naturally occurring ones. In this chapter, we shall have a look at a tool called GenERRate.[23]

In order for the synthetic error corpus to be useful in an error detection system, the errors that are introduced need to resemble those that the system aims to detect. Thus, the process is not without some manual effort: knowing what kind of errors to introduce requires the inspection of real error data, a process similar to error annotation. Once the error types have been specified, though, the process is fully automatic and allows large error corpora to be generated. If the set of well-formed sentences into which the errors are introduced is large and varied enough, it is possible that this will result in ungrammatical sentence structures which learners produce but which have not yet been recorded in the smaller naturally occurring learner corpora. To put it another way, the same type of error will appear in lexically and syntactically varied *contexts*, which is potentially advantageous when training a classifier. (The procedure described in this chapter will result in a monolithic error corpus with no concept of different backgrounds, different genres and so on, although it would of course be possible to create a series of synthetic corpora with different characteristics insofar as an appropriate description can be given for each.)

[23]) The requirements of such a tool, in particular the types of error it should include, were discussed extensively with Jennifer Foster, and the implementation of GenERRate is mostly due to her.

5.1. Earlier use of artificial error data

Artificial errors have been employed previously in targeted error detection. Sjöbergh & Knutsson introduced split-compound errors and word-order errors (2005) into Swedish texts and used the resulting artificial data to train their error detection system. These two particular error types were chosen because they are frequent errors amongst non-native Swedish speakers whose first language does not contain compounds or has a fixed word order. They compared the resulting system to three Swedish grammar checkers, and found that their system had higher recall, for certain types of error almost twice as high as other systems, at the expense of lower precision when tested on a collection of errors, although it could not compete with state-of-the-art systems on more realistic data with lower error rates. Brockett, Dolan & Gamon introduced errors involving mass/count noun confusions into English news-wire text and then used the resulting parallel corpus to train a phrasal smt system to perform error correction (2006). This system was only tested on a collection of errors, and found to give much higher recall than Microsoft Word's system, again at the expense of lower precision. Lee & Seneff automatically introduced verb form errors (subject–verb agreement errors, complementation errors and errors in a main verb after an auxiliary) (2008*b*), parsed the resulting text and examined the parse trees produced. The 'disturbances' in the parse trees observed for the sentences into which errors had been introduced were then used as indicative of error.

Both Okanohara & Tsujii (2007) and Wagner, Foster & Genabith (2007) attempted to learn a model which discriminates between grammatical and ungrammatical sentences, and both used synthetic negative data obtained by distorting sentences from the bnc. The methods used to distort the bnc sentences are, however, quite different: Okanohara & Tsujii generated ill-formed sentences by sampling a probabilistic language model and ended up with 'pseudo-negative' examples which resemble machine translation output more than language produced by humans:

> We know of no program, and animated discussions about prospects for trade barriers or regulations on the rules of the game as a whole, and elements of decoration of this peanut-shaped to priorities tasks across both target countries.

Indeed, machine translation is one of the applications of their resulting discriminative language model, which is able to distinguish between correct and incorrect sentences with an accuracy of 74 per cent (given a set of incorrect sentences consisting of this type of highly deviant constructions). Wagner, Foster & van Genabith introduced grammatical errors of the following four types into bnc sentences: context-sensitive spelling errors, agreement errors, errors involving a missing word, and errors involving an extra word. All four types were considered equally likely and the resulting synthetic corpus contains errors that look like the kind of slips that would be made by native speakers (*e.g.*, repeated adjacent words) as well as errors that resemble learner errors (*e.g.*, missing articles). They reported a drop

in accuracy for their classification methods when applied to real learner texts as opposed to held-out synthetic test data (WAGNER, FOSTER & GENABITH 2009), reinforcing the earlier point that artificial errors need to be tailored for the task at hand.

Artificial error data has also been put into use in the automatic *evaluation* of error detection systems, as exemplified by BIGERT's use of a tool called Missplel to generate spelling errors used to evaluate a context-sensitive spelling checker (2004). Furthermore, the performance of general-purpose NLP tools such as part-of-speech taggers and parsers in the face of noisy ungrammatical data has been automatically evaluated using artificial error data. Since the features of machine-learned error detectors are often part-of-speech *n*-grams or word–word dependencies extracted from parser output (*cf., e.g.,* DE FELICE & PULMAN 2008), it is important to understand how part-of-speech taggers and parsers react to particular grammatical errors. BIGERT *& al.* introduced artificial context-sensitive spelling errors into error-free Swedish text and then evaluated parsers and a part-of-speech tagger on this text using their performance on the error-free text as a point of reference (2005). Similarly, FOSTER investigated the effect of common English grammatical errors on two widely-used statistical parsers by means of distorted treebank trees (2007), using the procedure described by WAGNER, FOSTER & GENABITH to introduce errors into the treebank sentences (2007, 2009).

Finally, negative evidence in the form of automatically distorted sentences has been used in unsupervised learning. SMITH & EISNER generated negative evidence for their *contrastive estimation* method by moving or removing a word in a sentence (2005*a*, 2005*b*). Since the aim of this work was not to detect grammatical errors, there was no requirement to generate the kind of negative evidence that might actually be produced by either native or non-native speakers of a language. The negative examples were used to guide the unsupervised learning of a part-of-speech tagger and a dependency grammar.

We can conclude from this survey that synthetic error data has been used in a variety of NLP applications, including error detection and evaluation of error detectors. In the next section, we describe an automatic error generation tool, which has a modular design and is flexible enough to accommodate the generation of the various types of synthetic data described above.

5.2. Error generation tool

GenERRate is an error generation tool which accepts as input a corpus and an error analysis file consisting of a list of error types and produces an error-tagged corpus of syntactically ill-formed sentences. The sentences in the input corpus are assumed to be grammatically well-formed. GenERRate is implemented in Java and is available as a download for use by

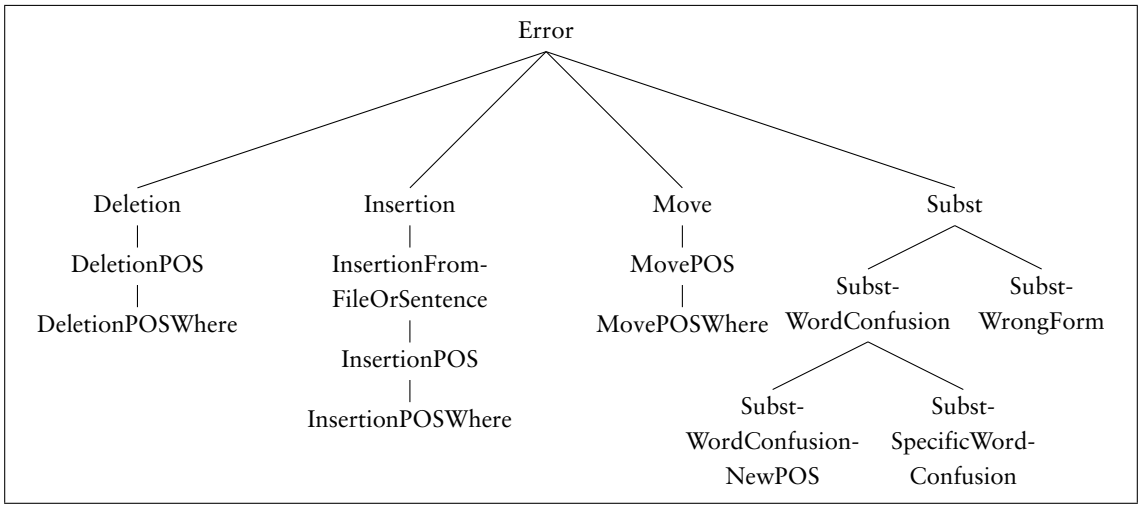

FIGURE 5.1. GenERRate error types.

other researchers.[24]

5.2.1. Supported error types

Error types are defined in terms of their deviance, that is, in terms of the operations (*deletion, insertion, move* and *substitution*) which are applied to a well-formed sentence to make it ill-formed and which can be reversed to correct the error. As well as being a popular classification scheme in the field of error analysis (JAMES 1998), it has the advantage of being theory-neutral. This is important in this context since it is hoped that GenERRate will be used to create negative evidence of various types, be it learner-like grammatical errors, native-speaker slips or more random syntactic noise. GenERRate is intended to be easy to use for anyone working in linguistics, applied linguistics, language teaching or computational linguistics.

The inheritance hierarchy in Fig. 5.1 shows the error types that are supported by GenERRate. The error types can be briefly described as follows:

1. Errors generated by removing a word

   a) *DeletionError:* Remove a randomly selected word.
   b) *DeletionPOSError:* Extends DeletionError by allowing a specific part of speech to be specified.
   c) *DeletionPOSWhereError:* Extends DeletionPOSError by allowing left and/or right context to be specified (part-of-speech tag or *start/end* to indicate beginning/end of sentence).

[24]) http://www.computing.dcu.ie/~jfoster/resources/genERRate.html

2. Errors generated by inserting a word

    a) *InsertionError:* Insert a randomly chosen word at a random position in the sentence. The word is chosen either from the sentence itself or from a word list, and this choice is also made at random.

    b) *InsertionFromFileOrSentenceError:* Extends InsertionError by allowing to specify whether to choose a word from the sentence or from a word list.

    c) *InsertionPOSError:* Extends InsertionFromFileOrSentenceError by allowing the part of speech of the inserted word to be specified.

    d) *InsertionPOSWhereError:* Extends InsertionPOSError by allowing left and/or right context to be specified.

3. Errors generated by moving a word

    a) *MoveError:* Move a randomly selected word to a random position.

    b) *MovePOSError:* Extends MoveError by allowing a specific part of speech to be specified.

    c) *MovePOSWhereError:* Extends MovePOSError by allowing the change in position to be specified in terms of direction and number of words.

4. Errors generated by substituting a word

    a) *SubstError:* Replace a random word by a word chosen at random from a word list.

    b) *SubstWordConfusionError:* Extends SubstError by allowing the part of speech to be specified (same part of speech for both words).

    c) *SubstWordConfusionNewPOSError:* Extends SubstWordConfusionError by allowing two different parts of speech to be specified (*i.e.*, a word of one part of speech replaces a word of another part of speech, both explicitly indicated).

    d) *SubstSpecificWordConfusionError:* Extends SubstWordConfusionError by allowing specific words to specified (*e.g.*, the substitution of *be* for *have*).

    e) *SubstWrongFormError:* Extends SubstError by allowing a substitution with a different form of the same word to be specified. Currently implemented substitutions include noun number (*word/words*), verb number (*write/writes*), tense (*write/wrote*), adjective form (*big/bigger*) and adjective/adverb confusion (*quick/quickly*). Note that only this error type would require adaptation to be useful for another language (at the moment, only English morphology is supported).

5.2.2. Input corpus

The corpus of well-formed sentences that is supplied as input to GenERRate must be split into sentences. It does not have to be part-of-speech tagged, but it will not be possible to generate many of the errors if it is not. GenERRate has been tested using two part-of-speech tagsets: the Penn Treebank tagset (SANTORINI 1990) and the CLAWS tagset (GARSIDE 1987; *see* Appendix C).

The error analysis file specifies the errors that GenERRate should attempt to insert into the sentences in the input corpus. A toy example with the Penn tagset might look like the following:

```
subst,word,an,a,0.2
subst,NNS,NN,0.4
subst,VBG,TO,0.2
delete,DT,0.1
move,RB,left,1,0.1
```

The first line is an instance of a SubstSpecificWordConfusionError, the second and third are instances of the SubstWrongFormError type, the fourth is a DeletionPOSError, and the fifth is a MovePOSWhereError. The number in the final column specifies the desired proportion of the particular error type in the output corpus and is optional; however, if it is present for one error type, it must be present for all. The overall size of the output corpus is supplied as a parameter when running GenERRate.

5.2.3. Error generation

When frequency information is not supplied in the error analysis file, GenERRate iterates through each error in the error analysis file and each sentence in the input corpus, tries to insert an error of this type into the sentence and writes the resulting sentence to the output file together with a description of the error. GenERRate includes an option to write the sentences into which an error could not be inserted and the reason for the failure to a log file. When the error analysis file *does* include frequency information, a slightly different algorithm is used: for each error, GenERRate selects sentences at random from the input file and attempts to generate an instance of that error until the desired number of errors has been produced or the set of input sentences has been exhausted.

5.3. Classification experiments

In our original GenERRate paper, results from two binary classification experiments using synthetic training data were reported (FOSTER & ANDERSEN 2009). Our aim was not so

CLC	GenERRate
<W>*available*$_{JJ}$ *position*|*position available*$_{JJ}$</W>	move JJ
<RJ>*unknown*$_{JJ}$|*strange*$_{JJ}$</RJ>	subst JJ
<RT>*of*$_{IO}$|*for*$_{IF}$</RT>	subst IF IO
<RN>*house*$_{NNL1}$|*building*$_{NN1}$</RN>	subst NN1 NNL1
recommend$_{VVO}$ <UA>*you*$_{PPY}$|</UA> *the*$_{AT}$	insert file VV0 PPY AT
with$_{IW}$ <MD>|*an*$_{AT1}$</MD> *apology*$_{NN1}$	delete IW AT1 NN1
<FN>*complaint*$_{NN1}$|*complaints*$_{NN2}$</FN>	subst NN2 NN1
<FV>*to*$_{TO}$ *understand*$_{VVO}$|*understanding*$_{VVG}$</FV>	subst VVG TO
<AGN>*brain*$_{NN1}$|*brains*$_{NN2}$</AGN>	subst NN2 NN1
<AGD>*that*|*those*</AGD>	subst word those that
<AGV>*is*|*are*</AGV>	subst word are is
<DY>*proper*$_{JJ}$|*properly*$_{RR}$</DY>	subst RR JJ
<DA>*my*|*mine*</DA>	subst word mine my

FIGURE 5.2. Examples of errors from the CLC with corresponding GenERRate error descriptions. The upper part shows word-order errors, insertions, omissions and lexical replacements, whereas morphological errors are shown in the lower part.

much to improve classification performance as to test the GenERRate tool, to demonstrate how it can be used, and to investigate differences between synthetic and naturally occurring datasets. Whereas synthetic errors generated by our tool were successfully shown to be superior to (*i.e.*, closer to real errors than) previously used synthetic errors and thus more useful as training data, our attempt to create a large error corpus inspired by the CLC and use it in a binary sentence classification task, similar to the one described in the previous chapter, was less conclusive. This section describes this experiment in detail.

5.3.1. Setup

We attempted to use GenERRate to insert errors into corrected CLC sentences. In order to do this, we needed to create a CLC-specific error analysis file, which was done automatically by extracting erroneous part-of-speech trigrams from the error-annotated CLC sentences and encoding them as GenERRate errors (*cf.* Fig. 5.2). This resulted in approximately 13,000 errors of the following types: DeletionPOSWhereError, InsertionPOSWhereError, MovePOSError, SubstWordConfusionError, SubstWordConfusionNewPOSError, SubstSpecificWordConfusionError and SubstWrongFormError. Frequencies were extracted, and errors occurring only once were excluded.

We trained a series of naïve Bayesian classifiers using different combinations of real and synthetic errors in different quantities, including the following three classifiers: The first was trained on corrected CLC sentences (the grammatical section of the training set) and original CLC sentences (the ungrammatical section). The second classifier was trained on corrected CLC sentences and the sentences generated from the corrected CLC sentences using GenERRate (we call these 'faux-CLC'). The third was trained on corrected CLC sentences

Feature	Examples
Word	*a; thought; occur+ed*
Word bigram	*a thought; thought occur+ed*
Part of speech	ATI; NNI; VVD
Part-of-speech bigram	ATI NNI; NNI VVD
Part-of-speech trigram	ATI NNI VVD

FIGURE 5.3. Overview of the features used. The examples are taken from the fragment a_{ATI} $thought_{\text{NNI}}$ $occurred_{\text{VVD}}$.

Training data	Correct/incorrect		Correct/faux		Correct/incorrect+faux	
Confusion matrix	Correct	Incorrect	Correct	Incorrect	Correct	Incorrect
Correct sentences	34,814	8,825	35,935	7,704	36,110	7,529
Incorrect sentences	26,456	18,917	33,233	12,140	28,058	17,315
Precision	69.7%		62.0%		69.7%	
Recall	42.6%		30.7%		38.2%	
Accuracy	61.3%		55.1%		60.0%	

FIGURE 5.4. Classification performance on the test data for different training sets.

and a 50/50 combination of CLC and faux-CLC sentences. In all experiments, the grammatical section of the training data contained 438,150 sentences and the ungrammatical section 454,337. The classifiers were tested on a held-out section of the CLC containing 43,639 corrected CLC sentences and 45,373 original CLC sentences. To train the classifiers, the MALLET implementation of naïve Bayes was used.We used a subset of the features mentioned in the previous chapter (*see* Fig. 4.1), namely word unigrams and bigrams, as well as part-of-speech unigrams, bigrams and trigrams, illustrated in Fig. 5.3; these are features that can be constructed directly from the faux-CLC sentences as output by GenERRate, thus avoiding the need for parsing.

5.3.2. Results

The results of the CLC classification experiment are presented in Fig. 5.4. There is a 6.2 per cent drop in accuracy when we move from training on original CLC sentences to artificially generated sentences, which is somewhat disappointing since it means that we have not completely succeeded in replicating the CLC errors using GenERRate. Most of the accuracy drop is on the ungrammatical side: the correct/faux model classifies more incorrect CLC sentences as correct than the correct/incorrect model. One reason for this is that certain frequent types of error are not included in the error analysis file and cannot be described within the current framework which largely relies on part-of-speech tags: the corrected CLC sentences used to generate the faux-CLC set were tagged with the CLAWS tagset, and although more fine-grained than the Penn tagset, it does not, for example, make a distinction between mass and count nouns, a common source of error. Another important reason for the drop in accuracy is the recurrent spelling errors which occur in the incorrect CLC test

set but not in the faux-CLC test set. It is promising, however, that much of the performance degradation is recovered when a mixture of the two types of ungrammatical training data is used, suggesting that artificial data could be used to augment naturally occurring training sets. Unfortunately, faux-CLC data actually gives a small decrease in performance when added to the full set of incorrect sentences found in the CLC. It would be interesting to see whether certain types of artificially generated errors are responsible for this deterioration, whereas others might perhaps be beneficial; furthermore, additional synthetic training data may be more useful in combination with a more extensive feature set, in which case the data sparsity problem will be more acute.

5.4. Limitations of GenERRate

The following three limitations, which became apparent to us during the classification experiments, illustrate some of the issues that make the task of generating synthetic error data non-trivial.

5.4.1. Sophistication of the error specification

There are some coverage issues with GenERRate, some of which are due to the simplicity of the supported error types. When linguistic context is supplied for deletion or insertion errors, it takes the form of the part of speech of the words immediately to the left and/or right of the target word. LEE & SENEFF analysed preposition errors made by Japanese learners of English (2008*a*) and found that a greater proportion of errors in argument prepositional phrases (*look at him*) involved a deletion than those in adjunct prepositional phrases (*came at night*). The only way for such a distinction to be encoded in a GenERRate error analysis file is to allow *parsed* input to be accepted. This brings with it the problem that parsers are less accurate than part-of-speech taggers, but it may still be preferable to let the system make a certain number of mistakes than not to be able to specify the conditions for error insertion because the representation is inadequate. A more significant improvement would be to make use of WordNet synonym sets or another source of semantic similarity in order to choose the new word in substitution errors, knowing that humans hardly ever substitute one word for another without there being some sort of connection between them.

5.4.2. Covert errors

A covert error is an error that results in a syntactically well-formed sentence with an interpretation different from the intended one. Covert errors are a natural phenomenon, occurring in real corpora. LEE & SENEFF gave the example *I am preparing for the exam,* which has been annotated as erroneous because, given its context, it is clear that the person

meant to write *I am prepared for the exam* (2008*b*). The problems lie in deciding what covert errors should be handled by an error detection system and how to create synthetic data which gets the balance right.

Covert errors can be produced by GenERRate as a result of the sparse linguistic context provided for an error in the error analysis file. An inspection of the generated errors shows that some error types are more likely to result in covert errors. An example is the SubstWrongFormError when it is used to change a noun from singular to plural. This results in the sentence *But there was no sign of Benny's father* being changed to the well-formed but more implausible *But there was no sign of Benny's fathers*. The next version of GenERRate should include the option to change the form of a word only when it appears in a certain context.

In the design of GenERRate, particularly in the design of the SubstWrongFormError type, the decision was made to exclude tense errors because they are likely to result in covert errors (*e.g., she walked/walks home*), but in doing so we also avoid generating examples like this one:

(15) *When I was a high school student, I *go/went to bed at one o'clock.*

These tense errors are common in learner data and their omission from the faux-CLC training set is one of the reasons why the performance of this model is inferior to the real-CLC model.

5.4.3. More complex errors

The learner corpora contain some errors that are corrected by applying more than one transformation. Some are handled by the SubstWrongFormError type (*I spend a long time *to fish / fishing*), but some are not (*she is one of *reason / the reasons I became interested in English*).

5.5. Conclusion and further work

We have presented GenERRate, a tool for automatically introducing syntactic errors into sentences and shown how it can be used to create synthetic training data for grammatical error detection research. Although we have focused on the binary classification task, we also intend to test GenERRate in targeted error detection. Another avenue for future work is to explore whether GenERRate could be of use in the automatic generation of language test items (*cf., e.g.,* CHEN, HSIEN-CHIN & CHANG 2006). Our immediate aim as far as GenERRate development is concerned is to produce a new version which tackles some of the coverage issues highlighted by our experiments.

Although GenERRate can provide training data that is more useful than previous collections of synthetic errors (FOSTER & ANDERSEN 2009), it does still not rival a real error corpus. More experiments are needed to see whether specific types of synthetic errors can constitute a useful supplementary source of incorrect examples, perhaps with the addition of more sophisticated error generation mechanisms as suggested previously.

Finally, it should be mentioned that the lack of a generally available error corpus for use as test data acts as a significant impediment to the comparison of different systems' respective performance. A synthetic test corpus would however have to be closely modelled on the errors and error distribution in a real corpus lest it constitute an artificial yardstick of little relevance to the actual task, and thus necessarily reveal some essential characteristics of the model corpus; publishing a small part of the CLC (or a similar corpus) might be a better solution.

Chapter 6.

Replacement errors

Data sparsity is a problem that permeates all empirical or corpus-based approaches to natural language processing, and it exhibits itself quite prominently in the task of error detection, since perfectly correct but for some reason unusual sentences, including the ones a more or less specialised language generation system may choose to avoid, should not be flagged as incorrect just because they happen to be statistically improbable. In this chapter, we shall see how data from the bnc combined with WordNet and clustering methods can provide the necessary evidence for detecting two types of lexical replacement errors, namely adjectival and prepositional choice errors.

6.1. Adjectival choice errors

Unlike glaring syntax errors like failing verb–noun agreement, adjectival choice issues must typically be ascribed to lack of idiomaticity rather than infringement of a clear grammatical rule. This suggests that the information required to detect infelicitous adjective–noun pairs can be extracted directly from a reference corpus, at least to the extent to which the native speaker's appreciation of idioms correlates with actual usage. One way of doing this, as well as the results obtained for a small set of adjectives, will be described in the following.

We posit that an inappropriately chosen adjective tends to be semantically close to or at least related to the one that should have been chosen. For instance, *brisk walk* is probably more idiomatic in some sense than *nice walk*, but we would not want to suggest the former as a replacement for the latter since they mean quite different things.

On the other hand, *fast walk* is mostly synonymous with — and could, perhaps advantageously, be replaced by — *brisk walk*. Proposing more idiomatic alternatives to perfectly acceptable expressions can be useful for some applications, but not if our aim is to detect ungrammatical constructions proper. Getting this right is at least in part a question of appropriate thresholds.

	great	144	†	long	6		extensive	2		huge	
	large	122		heavy	5		vast			greatest	
	wide	33	†	terrible	4	†	sufficient		†	grand	
†	high	14	†	tall	4		significant		†	favourite	
	broad	10	†	major	4	†	serious		†	best	
	loud	9		tremendous	3	†	popular			bad	
†	good	8	†	considerable	3		important				

FIGURE 6.1. Adjectives replacing *big* in the CLC, sorted by frequency (1 not explicitly indicated). The ones that do not belong to the same synonym set as *big* in WordNet are marked with a dagger.

We make the assumption that all 'significant' idiomatic adjective–noun combinations occur 'frequently' in a big corpus like the British National Corpus (BNC). This means in particular that any adjective will be considered appropriate in conjunction with a rare noun and, furthermore, that any adjective in a set of confusables, typically quasi-synonyms, likewise will be considered appropriate if none of them co-occur 'frequently' with a given noun, since there is no clear evidence either way.

In the following, we shall investigate the feasibility of automatically detecting infelicitous instances of the adjective *big* found in the Cambridge Learner Corpus (CLC). This happens to be the most commonly misused adjective in the corpus (383 errors), and the class of magnitude adjectives has also been discussed previously (COPESTAKE 2005).

6.1.1. Semantic relatedness

Fig. 6.1 shows the full set of adjectives suggested as adequate replacements for *big* in different contexts. As one might expect, a small set of fairly common adjectives accounts for a large proportion of the corrections, and most of these are indeed semantically close to *big*. WordNet's synonym set confirms our intuition: it may well be that only 14 out of 27 words appear in the same set as *big*, but these cover more than 94 per cent of the instances. This suggests that synonym sets like the ones defined by WordNet might be used to approximate real confusion sets, which will typically be useful for less common adjectives.

In 319 of the 383 instances (83%), the parser successfully identifies the adjective as directly modifying a noun. As can be seen from the examples below, the correction provided by the annotator tends to keep the idea of magnitude, though this is less obvious when the appropriate scale is qualitative, as in the last two examples.

(16) a. *a *big/large number of tourists*

b. *beautiful views and a *big/wide range of facilities*

c. *a *big/broad experience of working with children*

d. *cause *big/tremendous disasters we can't imagine*

big	*	grand		long		tall
broad		heavy		loud		terrible
considerable		high	**	main		tremendous
extensive	*	huge		major		wide
good	*	important	**	numerous		
great		large	**	strong		

FIGURE 6.2. Confusion set for *big*. This set includes, apart from the word itself, all the adjectives that are proposed at least twice as a replacement for *big* in the CLC, as well as some quasi-synonyms proposed only once (marked with an asterisk) or not at all (marked with a double asterisk). The choice of a confusion set which does not reflect the CLC perfectly is intended to make the task of selecting the correct word more realistic.

e. *a *big/wide variety of hot meals every day*

f. *quite friendly with a *big/good sense of humour*

g. *My *big/favourite hobby is horse-riding.*

In the following experiments, our test set comprises these 319 occurrences of *big* represented in terms of incorrect adjective *(i.e., *big),* correct adjective proposed by the annotator and noun being modified *(e.g., *big, large, number)*.

6.1.2. Corpus frequencies vs. annotator judgements

Co-occurrence statistics extracted from the BNC can be used to determine whether the annotators' preferences in cases where *big* has been marked up as erroneous agree with evidence of usage: We used the Robust Accurate Statistical Parsing system (RASP) to identify adjective–noun relations in the BNC and considered the set of adjectives listed in Fig. 6.2 as semantically close to and thus potentially confusable with *big*. We then ranked the adjectives in the confusion set, for each of the nouns in the test set, according to the number of times each adjective co-occur with the noun in question. (Unlike more ambitious collocation metrics, the frequency/rarity of each word separately is not taken into account, which effectively gives a bias towards more common adjectives in our case.)

The naïve strategy of choosing the most common adjective for each noun gave the following result when applied to the test set of 319 instances of inappropriate use of *big:* in 223 cases, the most common adjective in the BNC is also the one chosen by the annotators; in 88 cases, the annotators and the BNC differ (sometimes because the annotators have chosen a correction outside our confusion set), but neither ends up with *big;* in 6 cases, no evidence can be found as the noun does not co-occur with any of the adjectives in the confusion set; and in the remaining 2 cases, *big* actually turns out to be the most commonly chosen adjective. Obviously, there will often be more than one possible correction, so the fact that the BNC and the annotators sometimes suggest different ones is not necessarily a problem.

6.1.3. Error detection and correction

In a more realistic scenario, the system will have to identify incorrect instances of *big* amidst a considerably larger number of correct ones. (Even in the CLC, the adjective is actually used correctly 95 per cent of the times it occurs.) By using the method previously applied to incorrect instances, we were able to extract 5,810 purportedly correct adjective–noun pairs. We should then like the system to detect as many of the 319 incorrect instances as possible without mistakenly condemning the correct ones.

The BNC provides a good account of correct English usage, but should not be taken as an infallible gold standard. In particular, its breadth of coverage inevitably leads to inclusion of marginal usage, which in our case implies that an adjective–noun combination which occurs only twice or thrice in the BNC should not necessarily be regarded as a strong collocation. Perhaps even more importantly, a sizeable difference in frequency is required for one adjective to be considered as clearly 'better' than another in a given context. (Excluding certain genres, such as poetry and unscripted speech, can be expected to filter out many expressions that are not really part of the standard language, but this approach was not tested.)

We started with a rather conservative threshold, considering an adjective as a possible alternative to *big* only when the adjective–noun combination occurs at least 100 times more frequently in the BNC than the corresponding *big*–noun combination, or at least 100 times if *big*–noun cannot be found at all. Choosing the most frequent adjective in each case then permitted 90 out of the 319 incorrect occurrences of *big* to be correctly identified as such (28% recall), and the system provided the same correction as the annotator in 79 cases. On the other hand, 25 of the correct occurrences were also signalled as incorrect, including the following examples:

(17) a. **big/wide range*
b. **big/large number*
c. **big/wide variety*

As the attentive reader will have noticed, these corrections are identical to those provided by the annotators for the same constructions occurring elsewhere in the corpus (*cf.* Ex. 16), and our preliminary conclusion is that most, if not all, of these cases really should be regarded as incorrect, and that mere oversight has led to their not being annotated accordingly.

Lowering the threshold to 50 allowed the system to detect 152 errors (48% recall) and correct 132 of them in accordance with the annotators' prescription. An additional 27 correct instances (52 in total) were also classified as incorrect. It would seem that most of these really ought to be corrected, but the system may be led astray in contexts where two adjectives taken from our set of confusables are not actually synonymous:

(18) **big/good fortune*

The topic under discussion is monetary wealth, which means that the proposed correction rather distorts the meaning. Whether or not the original is actually infelicitous may be a matter of opinion. It is interesting to note, however, that the expression *large fortune* does occur in the BNC, albeit much less frequently than *good fortune,* whereas *big fortune* cannot be found at all.

With an even more liberal threshold of 25, the system found 187 errors (59% recall) and corrected 160 successfully, whereas the number of correct occurrences identified as incorrect rose to 155, including at least a few which should clearly not be considered as incorrect:

(19) **big/great deal*

As a matter of fact, *big deal* occurs 58 times in the BNC, which suggests that this problem could, at least in part, be alleviated by using a slightly more sophisticated metric, in particular not considering combinations that actually do occur a certain number of times in the BNC as potentially erroneous.

For this technique to be useful for practical applications, it would have to be extended to cover other classes of words in addition to the set of adjectives related to *big;* an immediate extension to the work presented here would be to check whether it applies to other adjectival confusion sets as well.

6.2. Prepositional choice errors

In the remaining part of this chapter, the particular problem of detecting prepositional choice errors in a syntactic verb–preposition–noun context will be investigated, using a model based on the intuition that a preposition is likely to be incorrect if there is overwhelming evidence for another one in the same context. We shall see how co-occurrence statistics for verb–preposition–noun triples can be accumulated over sets of verbs and nouns with similar prepositional preferences and thus allow the information present in a corpus to be exploited more effectively, which gives a significant increase in recall without loss in precision.

Prepositions fulfil a wide range of linguistic functions, and the choice of a particular preposition in a given context often appears to be governed by complex and elusive criteria which cannot easily be summarised in such a way as to offer useful guidance to learners of the language, or even compiled into a potentially large set of rules for use in natural language generation or machine translation. A useful (albeit not always clear-cut) distinction is the one between *lexical* and *functional* use:

(20) a. *The milk is in the refrigerator.*

b. *The cat is on the refrigerator.*

(21) a. *Kimberly relies on her husband.*

b. *Kimberly believes in his wife.*

In Ex. 20, the change of preposition reflects a change in location, whereas the choice of preposition in Ex. 21 depends entirely on the verb, the preposition itself being essentially devoid of intrinsic meaning. We make no attempt to distinguish between these two uses in any formal way, but our focusing on prepositions for which a verbal head and a nominal dependent can be identified is thought to give a certain bias towards functional use. The very same challenge regarding prepositional choice may of course exist in phrases where the head is adjectival (*Mrs Kimberley is reliant on her spouse*) or nominal (*Mr Kimberley's belief in his consort*), and the method described in the following can be generalised to cover such cases as well.

The last few years have seen a rising interest in automatic detection of prepositional choice errors alongside the long-standing research problem of preposition generation or 'guessing'. Recent relevant publications include De Felice & Pulman, who trained a classifier using contextual features such as parts of speech, grammatical relations and semantic categories (2007, 2009); Tetreault & Chodorow, who used a similar model, with fewer and simpler features, augmented with heuristic rules (2008); and Gamon *& al.*, who presented a system incorporating in addition a 5-gram language model (2008).

It would of course be possible to identify incorrectly chosen prepositions without necessarily being able to suggest an appropriate replacement; however, at least partly owing to the scarcity of negative examples,[25] the typical approach, and the one we adopt here, is first to find either the ideal preposition in a given context or a set of allowable prepositions and then use this information to gauge the appropriateness of the original. The two are thus intimately related.

6.3. Model

We used the British National Corpus (bnc) to represent correct use of prepositions. The use of a balanced corpus like the bnc should give a fairly general model without bias towards a specific genre. It could of course be useful to adapt it for a specific application, but our

[25]) Whereas specimens of correct usage can easily be extracted from standard corpora, prepositions known to be wrong in a given context can only be found in an annotated error corpus, and a complete list of inappropriate choices would clearly be an elusive goal.

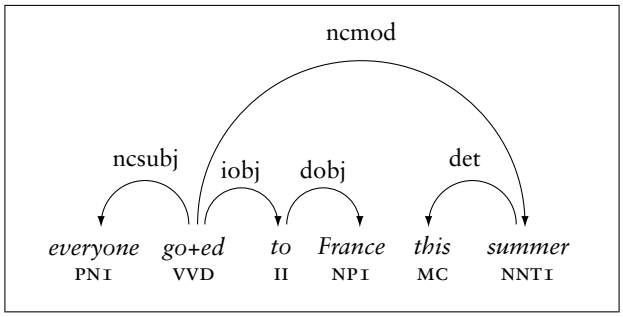

FIGURE 6.3. Analysis of the sentence *Everyone went to France this summer.* The words have been lemmatised, and the parser has added dependency relations (GRs). The arrows representing GRs are drawn pointing from head to dependent; the label above the arrow indicates the type of relation.

focus was on that which is incorrect across a wide variety of text types rather than that which might be unusual or inappropriate for a particular genre.

Once we have chosen a reference corpus, the next question is how to characterise the context of a preposition, given that we would like to predict prepositional choice based on the context in which the preposition appears.

(22) *Everyone went to France this summer.*

In Ex. 22, the choice of *to* is guided by the words *went* and *France*, which conveniently appear on either side of the preposition.

(23) *They all went together on the train.*

However, as Ex. 23 shows, immediate linear context is not always particularly helpful: *together* and *the* do not on their own provide particularly strong cues as to what the correct preposition is. Furthermore, we probably want to carry out lemmatisation in order to identify, for instance, the past tense *went* with the corresponding infinitive *go*.

To deal with these issues, we used a parsed version of the corpus and identified relevant features based on dependency relations. Fig. 6.3 shows how the sentence from Ex. 22 is analysed by the RASP system. Whenever there is an *iobj* relation from a verb to a preposition, and a *dobj* relation from the same preposition to a noun, a verb–preposition–noun triple (v, p, n) can be constructed. For instance, the sentence in Fig. 6.3 might give rise to the triple $(v, p, n) = (go, to, France)$.

The parser is not always successful at distinguishing between adjuncts and arguments, so it would arguably be better to include not only arguments (analysed as *iobj*), but also adjuncts (analysed as *ncmod*), especially since the choice of preposition may occasionally alter the analysis. However, the vast majority of prepositions that can be confidently identified as modifying a verb are analysed as arguments (95% in our set, based on only the most

probable analysis given by RASP for each sentence), so this should not have a significant impact on the results reported. Another solution, in particular if this had been more of a concern, would have been to consider the probabilistic GR set, incorporating the full set of possible analyses rather than the best parse only.

We were able to extract over a million distinct triples from the BNC, with frequencies varying from one to over a thousand. In the following sections, we use the notation $\nu(v, p, n)$ to refer to the number of occurrences of (v, p, n). The ultimate goal is to make use of this frequency information to detect erroneous use of prepositions.

6.4. Clustering

A naïve approach to error detection would be to consider a triple (v, p, n) to be correct if it occurs at least once in the BNC, and incorrect if it does not occur. This relies on the assumption that the BNC can be taken as a gold standard in the sense that no construction occurring even once should ever be considered as potentially incorrect, which may not be fully warranted given that the corpus includes slightly non-standard expressions as well as transcription errors and original printing errors. More importantly, $\nu(v, p, n)$ will often be low or zero even for correct constructions, which makes this approach inapplicable in its primitive form; in fact, as many as 40 per cent of the triples extracted from the BNC occur only once, which clearly suggests that there must be a large number of correct combinations, many of which cannot be expected to have been seen before, even in a large corpus.

Hence the idea of clustering: if we can ascertain that a noun n belongs to a cluster N of nouns which all behave in a similar fashion with respect to prepositional preferences, and that v likewise belongs to a cluster V of verbs behaving similarly to each other, the following accumulated frequencies can be calculated:

$$\nu(V, p, n) = \sum_{v' \in V} \nu(v', p, n)$$

$$\nu(v, p, N) = \sum_{n' \in N} \nu(v, p, n')$$

$$\nu(V, p, N) = \sum_{(v', n') \in V \times N} \sum \nu(v', p, n')$$

Problems of data sparsity can then be mitigated by using such generalised frequencies to augment or replace specific counts.

Rare words, namely verbs occurring in fewer than 10 triples and nouns occurring in fewer than 25 triples, were discarded, partly because a minimum of evidence is needed for the clustering to be meaningful, and partly as an efficiency/comprehensiveness trade-off. These thresholds gave 4,040 verbs and 5,858 nouns from the BNC to be clustered.

Obviously relevant features for verb clustering include prepositional co-occurrence counts $v(v, p_i) = \sum_n v(v, p_i, n)$ for each verb v, which can be derived directly from the (v, p, n) triples extracted from the BNC, and similarly $v(p_i, n) = \sum_v v(v, p_i, n)$ for noun clustering. The set of prepositions (p_i) used as co-occurrence features is fairly complete with 62 prepositions including *unto, opposite* and *aboard,* only exceedingly rare ones such as *vis-à-vis,* unabbreviated *versus* and elided *'fore* being excluded.

There are many other features of potential relevance, and we had initially planned to add more, but this small set of 62 features per verb/noun actually turned out to provide the information needed by the clustering algorithm to construct fairly good clusters.

6.4.1. Non-parametric Bayesian clustering

One problem with many clustering algorithms, which may have contributed to limiting their use in natural language processing applications, is that they typically require the number of clusters to be known in advance, which is often not possible in realistic scenarios. In contrast, non-parametric Bayesian models can discover a reasonable number of clusters based on the data whilst allowing the general level of granularity to be parameterised.

The following clustering experiments[26] are inspired by VLACHOS, KORHONEN & GHAHRAMANI, who used a Dirichlet process mixture model (DPMM) for clustering verbs into semantic classes using subcategorisation frames (2009). Each instance x_i with its characteristic features (in our case, each verb/noun with its characteristic prepositional distribution) is considered to have been generated by a distribution F with parameter θ_i,

$$x_i|\theta_i \sim F(\theta_i);$$

the parameters θ_i are in turn considered to derive from a distribution G,

$$\theta_i|G \sim G,$$

a Dirichlet process with base distribution G_0 and dispersion parameter α,

$$G|\alpha, G_0 \sim \mathrm{DP}(\alpha, G_0).$$

In other words, the instances x_i are drawn from a mixture of component distributions $F(\theta_i)$, and each component corresponds to a cluster.

The prior probability of assigning an instance to a particular cluster is proportionate to the number of instances already assigned to it; in other words, a DPMM exhibits the 'rich-get-richer' property. In addition, the probability that a new cluster be created is dependent on the dispersion parameter. A popular metaphor to describe a DPMM, which exhibits an

[26]) The actual clustering algorithm was implemented by Andreas Vlachos.

equivalent clustering property, is the Chinese restaurant process: customers (*sc.* instances) arrive at a Chinese restaurant with an infinite number of tables (*sc.* clusters); each customer sits down at one of the tables, previously occupied or vacant, with popular tables attracting more customers.[27]

The prior for assigning an instance x_i to either an existing component z or to a new one z' conditioned on the other component assignments (denoted by z_{-i}) is given by

$$p(z_i = z|z_{-i}) = \frac{n_{-i,z}}{N - 1 + \alpha}$$

$$\text{and} \quad p(z_i = z'|z_{-i}) = \frac{\alpha}{N - 1 + \alpha},$$

where N is the total number of instances, $n_{-i,z}$ is the number of instances assigned to component z excluding instance x_i, and α is the dispersion parameter. A clustering is generated by assigning more than one instance to the same mixture component. Instances that are assigned to the same component have equal θ_i's.

The distribution used to model the clusters is the multinomial (F), and the prior used is the Dirichlet distribution (G_0), which is the conjugate prior of the multinomial and therefore allows analytic integration over the parameters of the multinomial. Following Neal, the component assignments z_i of each instance x_i are sampled using the following scheme (2000):

$$P(z_i = z|z_{-i}, x_i) = b\, p(z_i = z|z_{-i})\, F(x_i|z_i = z, x_{-i,z}, \lambda),$$

where b is a normalising factor, λ are the parameters of the Dirichlet prior G_0, and $x_{-i,z}$ are the instances already assigned to component z (none if we are sampling the probability of assignment to a new component). This Gibbs sampling method is possible due to the fact that the instances in the model are interchangeable (*i.e.*, the order in which they are generated is not relevant); in terms of the Chinese restaurant process metaphor, we consider each instance x_i in turn as the last customer to arrive, and he chooses to sit together with other customers at an existing table or to sit at a new table. As did Navarro *& al.*, who used the same model to analyse variation between individuals (2006), we sampled the dispersion parameter α using the inverse Gamma distribution as a prior; only the parameters λ of the Dirichlet prior had to be set manually.

6.4.2. Evaluation issues

Unsupervised clustering is difficult to evaluate. Recent work has focused on evaluation of a clustering result given a manually created gold standard containing all the instances assigned to clusters according to the task at hand (Meilă 2007; Rosenberg & Hirschberg

[27]) The author is not familiar with any restaurant, Chinese or otherwise, whose customers behave in this way; the slightly different idea of already overcrowded *restaurants* that continuously attract new diners, whilst empty ones struggle to entice passers-by to enter, might be more evocative of the underlying concept.

2007). While using an external gold standard is a perfectly valid evaluation approach, it requires substantial human effort, especially if one considers that, unlike annotation of datasets for classification tasks, assignment of each instance to a cluster interacts with the assignments of other instances. Thus, in cases where the number of instances to be clustered is large, constructing a comprehensive gold standard for evaluation is usually not considered. Moreover, the very concept of a gold standard is problematic for 'unconstrained' clustering since there is no well-defined and predetermined set of categories.

A common way of assessing the utility of clustering as a processing step in the pipeline of a larger system is to look at the effect it has on the system's performance, a type of analysis we perform in the context of preposition guessing and correction (*see* Sect. 6.5); a more direct evaluation of the clustering quality is nevertheless desirable, not least because it allows the adequacy of the model (and the effect of different parameter settings) to be assessed at an earlier stage.

As we have seen, only purportedly correct instances from the BNC were used as input to the clustering algorithm, but those (even in the form of a held-out set) do not provide the basis for a good evaluation of the resulting clusters. Manually looking at some of the clusters may be sufficient to determine whether the clustering is potentially meaningful or minimally promising, but remains purely qualitative, and there is no such thing as a comprehensive gold standard for this task. As a partial solution, we made use of erroneous instances corrected in the CLC:

Consider a sentence like the following, where the candidate has written *in,* which the annotator has corrected to *to:*

(24) *Everyone went *in/to France this summer.*

Prepositional choice errors do not typically change the sentence structure[28], so we can proceed as outlined previously to extract the quadruple $(v_1, q, p, n) = (go, in, to, France)$, where q is the incorrect preposition and p is the correct preposition. Now, if we find the quadruple $(v_2, p, q, n) = (arrive, to, in, France)$, we know that v_1 (*arrive*) and v_2 (*go*) do not belong in the same cluster with respect to their prepositional preferences. Similarly, the quadruples $(v, q, p, n_1) = (sit, in, on, sofa)$ and $(v, p, q, n_2) = (sit, on, in, armchair)$ would tell us that the nouns n_1 (*sofa*) and n_2 (*armchair*) should not be clustered together.

Using this strong criterion, we can construct a set of verb pairs (and noun pairs) that should not be clustered together, and these pairs can be used to evaluate the results from the clustering experiments described in Sect. 6.4.3.

[28]) The parser component of the RASP system works on part-of-speech tags rather than words or lemmata, so it is typically unable to distinguish between prepositions. However, the prepositions *for, of* and *with* have unique tags, so the parse may change when one of these prepositions is involved.

One might reasonably be concerned about the validity of this evaluation, given that the rather strong criterion obviously precludes extensive coverage, whereas particular idioms may cause the clustering algorithm to be penalised unduly for clustering words which really behave similarly in most contexts (though it is not entirely obvious that words like *bed* and *futon* are as similar as one might at first expect). We argue that a less strict criterion could easily lead to an evaluation based on words which 'might reasonably' belong to different clusters or 'should probably not' be clustered together, which would not be particularly enlightening. In any case, this method actually gives a couple of thousand verb pairs (and noun pairs) upon which to base a preliminary evaluation, and we have found the results from this evaluation to correlate strongly with the performance obtained when the same clusters are used for preposition guessing and error detection in following experiments, and also agree roughly with our intuition of cluster quality as gauged by looking at cluster samples.

6.4.3. Clustering experiments

Feature choice and representation is crucial for unsupervised learning since, unlike in the case of supervised learning, there are no labelled instances that can provide guidance on correct interpretation and relative importance of individual features.

As already anticipated, preposition co-occurrence counts (*i.e.,* the number of times each preposition is used as a modifier of a given verb or the number of times each preposition is used with a given noun as its object) will be used as features in the following verb and noun clustering experiments. This choice is justified by our need to discover clusters of words that have similar behaviour with respect to preposition usage, which eventually can be used for preposition guessing and correction.

In the experiments reported below, we ran the Gibbs sampler 5 times, each time letting the sampler go through 100 iterations before we started sampling ('burn-in') in order to let the system reach a state that should not be affected by the initialisation, and then drawing 20 samples from each run with 5 iterations' lag between samples, values that have been reported to work well in prior work. The parameters λ of the symmetric Dirichlet prior was set to 10^{-4}, a very small value which allows for more flexible cluster formation, since at inference time the statistics of each new cluster generated is dependent almost exclusively on the features of the single instance assigned to it. Different values of λ were tried in preliminary experiments.

From the Gibbs sampler, we obtained a set of 100 samples in each experiment. As explained in Sect. 6.4, we intended to replace counts of single verbs/nouns with the counts of their respective containing clusters, which requires a unique set of clusters. While it would be desirable to average over the clustering samples obtained, this is not possible, since the clusters in a particular sample cannot be identified with any of the clusters in a different

sample. A different option would be to identify a particular sample as representative of all the samples, by measuring the average distance of each sample to all the others, but any individual sample is likely to contain clusters that are not representative of other samples.

To avoid these issues, we generated a unified clustering from the set of samples by means of the procedure used for qualitative evaluation by VLACHOS, KORHONEN & GHAHRAMANI (2009): we represented each clustering sample as a linking matrix and used the links that occur in a proportion greater than a given threshold *problink* of the samples to define the final clustering.

As an illustrative example, let us consider three clustering samples S_i of four elements x_i:

$$S_1 = \{\{x_1, x_2\}, \{x_3, x_4\}\} \qquad S_2 = \{\{x_1, x_3, x_4\}, \{x_2\}\} \qquad S_3 = \{\{x_1, x_4\}, \{x_2, x_3\}\}$$

A clustering sample S can be represented by a linking matrix $M = (m_{ij})$, where

$$m_{ij} = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ belong to the same cluster;} \quad \text{and} \\ 0, & \text{otherwise.} \end{cases}$$

The matrix M is symmetric by definition, and the elements m_{ii} do not contribute any useful information; we shall therefore represent it as an upper triangular matrix with dotted diagonal. This gives the following matrices:

$$M_1 = \begin{bmatrix} \cdot & 1 & 0 & 0 \\ & \cdot & 0 & 0 \\ & & \cdot & 1 \\ & & & \cdot \end{bmatrix} \qquad M_2 = \begin{bmatrix} \cdot & 0 & 1 & 1 \\ & \cdot & 0 & 0 \\ & & \cdot & 1 \\ & & & \cdot \end{bmatrix} \qquad M_3 = \begin{bmatrix} \cdot & 0 & 0 & 1 \\ & \cdot & 1 & 0 \\ & & \cdot & 0 \\ & & & \cdot \end{bmatrix}$$

Averaging over the linking matrices is trivial:

$$\bar{M} = \frac{1}{3} \sum_i M_i = \frac{1}{3} \begin{bmatrix} \cdot & 1 & 1 & 2 \\ & \cdot & 1 & 0 \\ & & \cdot & 2 \\ & & & \cdot \end{bmatrix}$$

The final clustering $\bar{S}$ is created by assigning all the pairs of elements x_i and x_j to the same cluster if the corresponding $\bar{m}_{ij}$ exceeds a threshold *problink*. Assuming that *problink* is set to ½ in our example, $m_{1,4}$ and $m_{3,4}$ exceed the threshold, which gives rise to the following clustering:

$$\bar{S} = \{\{x_1, x_3, x_4\}\}$$

The unified clustering thus obtained is partial, since instances which are clustered inconsistently in the set of samples will not be included. A lower *problink* threshold will in general give larger, less homogeneous clusters and increased coverage because of additional links between less similar instances. Note that this can either increase the number of clusters when instances were not linked otherwise, or decrease it when linking instances that already belong to other clusters.

	Triples		Verbs		Nouns	
In the CLC	36,447		36,447		36,447	
Found in the BNC	24,993	68.6%	36,413	99.9%	35,672	97.9%
Present in cluster input			36,370	99.8%	34,204	93.8%
In clusters (*problink* 0.75)			32,809	90.0%	31,503	86.4%
— (*problink* 0.85)			32,229	88.4%	29,337	80.5%
— (*problink* 0.99)			29,769	81.7%	23,963	65.7%

FIGURE 6.4. This table shows how many of the 36,447 triples extracted from the CLC can be found as complete triples in the BNC, as well as how many of them contain a verb/noun which can be found in one or more BNC triples, and furthermore how this verb/noun coverage changes as a result of filtering and clustering for different values of *problink*.

	Verb clusters		Noun clusters	
	Size	Wrong	Size	Wrong
WordNet	29.2	115	13.5	49
WordNet 1st sense	4.6	21	4.3	14
problink 0.75	13.2	4	37.8	42
problink 0.85	10.2	4	21.7	20
problink 0.99	6.1	2	8.7	13

FIGURE 6.5. Average cluster size and number of words clustered together which should not have been. Results are shown for different values of *problink*, and clusters constructed from WordNet synonyms are included for comparison.

6.4.4. Evaluation

As we have seen, the verb and noun clusters consist of words taken from the BNC, whereas we used data from the CLC for testing. Fig. 6.4 shows that almost all the verbs that appear in correct sentences in the CLC can be found in the BNC, and very few of them are filtered out before clustering because of low frequency; for nouns, the match between the CLC and the BNC is not quite as good, and a certain number of nouns get filtered out. Most of the shortfall, however, is caused by words which do not appear in the final clustering despite being given as input to the clustering algorithm, and this problem cannot be avoided completely since there are likely to be some words whose prepositional preferences do not correspond to any others' and which therefore cannot be clustered without loss in precision.

It is also worth noticing that almost one third of the (v, p, n) triples from the CLC are not found in the BNC, despite good coverage of both verbs and nouns separately.

As mentioned in Sect. 6.4.2, we extracted from the CLC a set of 2,100 pairs of verbs and 16,330 pairs of nouns which should not be clustered together, and we checked whether or not the words in each pair are assigned to different clusters for a given clustering. The results are summarised in Fig. 6.5 and show that our clusters do well according to this measure, whereas WordNet synonym sets violate the criterion more often by up to an order of magnitude for similar average cluster size.

For the rest of this chapter, results will be reported using the clusters we get with *problink*

0.85, chosen as a compromise of coverage, cluster size and accuracy. This clustering contains 287 verb clusters of size 2–85, and 255 noun clusters of size 2–233. Samples from some of the clusters are shown below:

Noun clusters
abolition, breeding, confirmation, erection, example, ...
academic, accountant, activist, admirer, american, pope, ...
act, charter, command, directive, grammar, law, ...
activity, agriculture, analysis, broadcast, census, ...
administration, bureau, bureaucracy, clan, company, ...
Africa, America, Anglia, Asia, Australia, Ayrshire, Baghdad, ...
aisle, alley, alleyway, corridor, glen, lane, meadow, ...
altitude, level.
apartment, ballroom, bar, barracks, bookshop, booth, ...
aquarium, ashtray, beirut, belfast, cemetery, gaol, oven, ...
contrary, grip, nuisance, toilet.
counter, desk.
discrimination, disruption, interference, loss.

Verb clusters
abandon, advertise, announce, revise, stipulate, stress, use.
abduct, actuate, administer, appropriate, back, betray, ...
abolish, cite, consider, diagnose, discover, doubt, enclose, ...
absorb, categorise, classify, infiltrate.
accuse, acquit, compose, conceive, convict, despair ...
achieve, attain, extinguish, manage, obey, reckon.
acquaint, associate, coincide, comply, cope, familiarise, ...
address, convey, explain, whittle.
adjourn, award, fine, offer, postpone, promise, refuse, ...
adjust, apply, suit, tailor.
alarm, appal, astonish, astound, dismay, distress, horrify, ...
annoy, bother, please.
ask, pity.

6.5. Experiments using the clusters

6.5.1. Preposition guessing

The task of choosing the/an appropriate preposition in a given context has been studied a bit more than that of preposition correction and is useful to look at for evaluation purposes, as well as to gain a better understanding of the issues involved. Given (v, p, n), where p is the original, correct preposition, we let the system choose the 'best' or most likely preposition p^* based on v and n only, *i.e.*,

$$p^* = \operatorname*{argmax}_{p'} \nu(v, p', n).$$

If $p^* = p$, the correct preposition has been chosen. Of course, several prepositions may be equally correct, potentially with different meanings, but this complication is typically not dealt with directly. Rather, human performance on the same task without further access to the context, or inter-annotator agreement, is used to establish an upper bound.

To avoid forcing a decision to be made when there is little evidence for any preposition, the system is allowed not to suggest a preposition:

$$p^* = \begin{cases} \operatorname*{argmax}_{p'} \nu(v, p', n), & \text{if} \quad \max_{p'} \nu(v, p', n) > \nu_0; \\ \text{none}, & \text{otherwise.} \end{cases}$$

Fig. 6.6 shows the results for different values of ν_0, and also when individual words v or n are replaced by the corresponding clusters V or N. As can be seen from the graph, performance decreases when cluster-based counts are used instead of those based on individual words. There is a bias against the clusters, however, since a word which does not belong to any cluster prevents the correct preposition from being chosen. When only words that appear in the final clustering are taken into account, as shown in Fig. 6.7, not only does the difference diminish, but verb clusters actually outperform individual words. The fact that clusters can replace individual words without a large decrease in precision on this task lends support to the idea that prepositional choice is not completely idiosyncratic, as it is clearly possible to group words together. The best model could well be one where high-frequency items are kept separate, whereas medium-frequency and low-frequency items can be clustered together. An analysis of which words are clustered and which are not would be a useful extension to this research.

Counts based on individual words and on clusters can also be combined: using clusters as back-off would give something like the following:

$$p^* = \begin{cases} \operatorname*{argmax}_{p'} \nu(v, p', n), & \text{if} \quad \max_{p'} \nu(v, p', n) > \nu_1; \\ \operatorname*{argmax}_{p'} \nu(V, p', n), & \text{otherwise, if} \quad \max_{p'} \nu(V, p', n) > \nu_2; \\ \operatorname*{argmax}_{p'} \nu(v, p', N), & \text{otherwise, if} \quad \max_{p'} \nu(v, p', N) > \nu_3; \\ \operatorname*{argmax}_{p'} \nu(V, p', N), & \text{otherwise, if} \quad \max_{p'} \nu(V, p', N) > \nu_4; \\ \text{none}, & \text{otherwise.} \end{cases}$$

A problem with this particular formulation is that it is difficult to set the thresholds ν_i; the only clear intution is that ν_1, which corresponds to individual words, should probably be set to a lower value than for instance ν_2, which corresponds to accumulated counts over verb clusters, but the ideal value would then depend on the cluster size. The identity

$$\operatorname*{argmax}_p \nu(v, p, n) = \operatorname*{argmax}_p \frac{\nu(v, p, n)}{\sum_p \nu(v, p, n)} = \operatorname*{argmax}_p \pi(p|v, n)$$

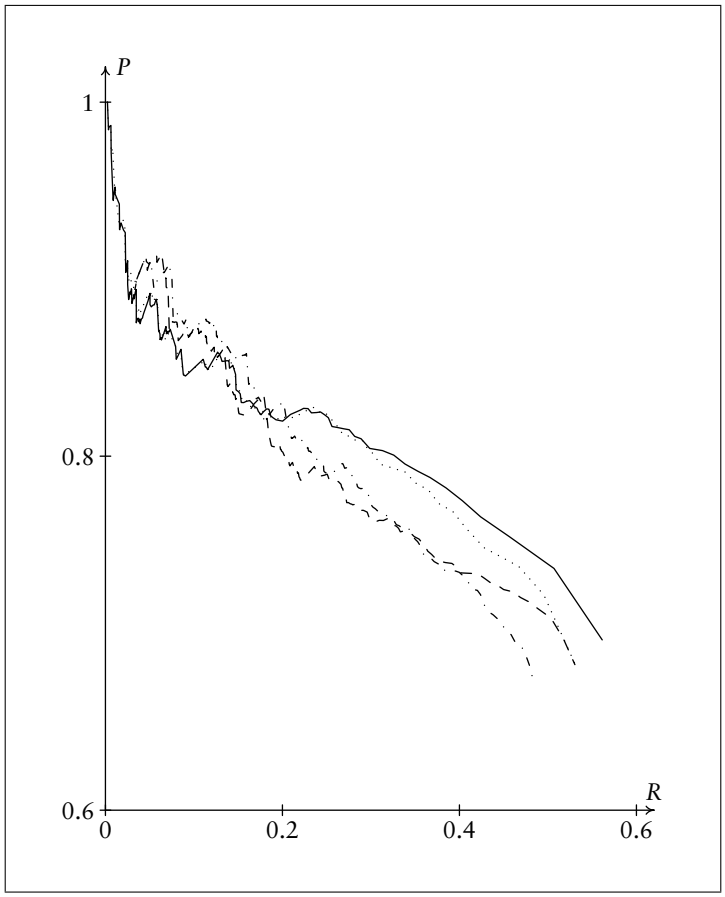

FIGURE 6.6. Precision–recall curve for preposition guessing, choosing the most frequent preposition based on individual words (continuous line), verb clusters (dotted), noun clusters (dashed) and both (dotted/dashed). The test set consists of originally correct (not corrected) instances of prepositions from the CLC.

shows that the most likely preposition p^* in a given context can be expressed equivalently in terms of probabilities π in place of raw counts ν. The back-off approach can then be defined as

$$p^* = \begin{cases} \operatorname{argmax}_{p'} \pi(p'|v,n), & \text{if} \quad \max_{p'} \pi(p'|v,n) > \pi_1; \\ \operatorname{argmax}_{p'} \pi(p'|V,n), & \text{otherwise, if} \quad \max_{p'} \pi(p'|V,n) > \pi_2; \\ \operatorname{argmax}_{p'} \pi(p'|v,N), & \text{otherwise, if} \quad \max_{p'} \pi(p'|v,N) > \pi_3; \\ \operatorname{argmax}_{p'} \pi(p'|V,N), & \text{otherwise, if} \quad \max_{p'} \pi(p'|V,N) > \pi_4; \\ \text{none}, & \text{otherwise.} \end{cases}$$

Unlike the unbounded thresholds ν_i, the thresholds π_i are probabilities in the range $[0, 1]$ and can, as a first approximation, be kept equal ($\pi_1 = \pi_2 = \pi_3 = \pi_4$) and adjusted en bloc for different precision/recall trade-offs.

An alternative approach is to combine the individual and cluster probabilities to a single

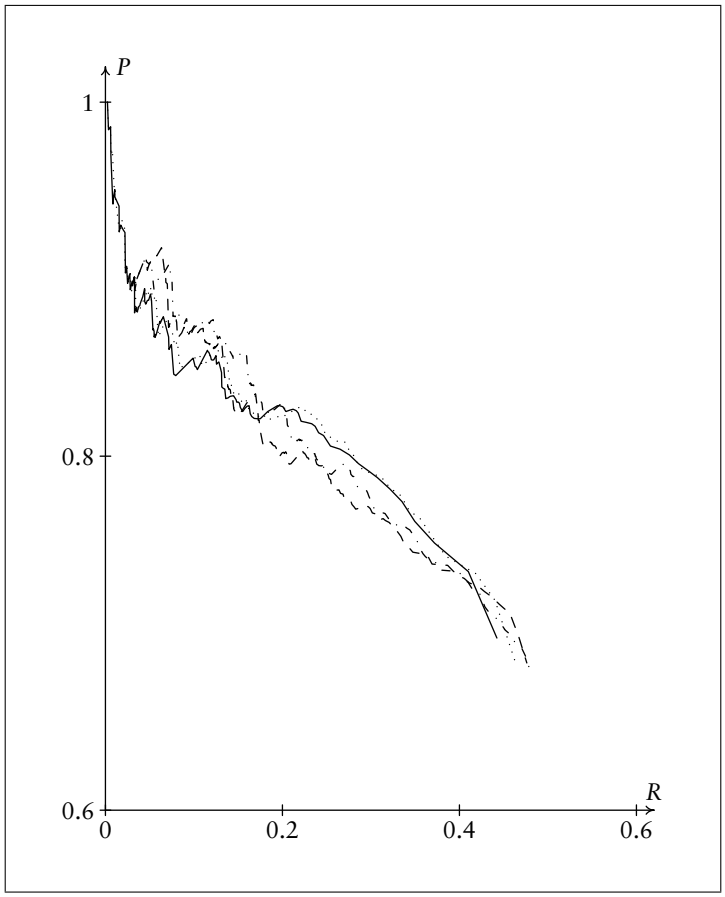

FIGURE 6.7. Same experiment as in Fig. 6.6, but the triples which contain either a verb or a noun that has not been clustered have been filtered out.

value,

$$\pi(p|v \in V, n \in N) = \lambda_1\pi(p|v,n) + \lambda_2\pi(p|V,n) + \lambda_3\pi(p|v,N) + \lambda_4\pi(p|V,N),$$

with $\lambda_i \geqslant 0$, $\sum_i \lambda_i = 1$, and define the best preposition as

$$p^* = \begin{cases} \operatorname{argmax}_{p'} \pi(p|v \in V, n \in N), & \text{if} \quad \max_{p'} \pi(p|v \in V, n \in N) > \pi_0; \\ \text{none}, & \text{otherwise.} \end{cases}$$

This allows the decision to be based on a combination of word-specific and more general selection criteria, but at the same time requires evidence to be available at different levels (except perhaps in the case of a low threshold π_0). Setting the coefficients λ_i to equal values ($\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/4$) is a reasonable starting point and gives good results compared to more carefully chosen values (*see* Fig. 6.8).

λ_1	λ_2	λ_3	λ_4	Correct	Incorrect	Accuracy
1	0	0	0	34,058	50,189	40.4%
0	1	0	0	38,320	45,927	45.5%
1/17	16/17	0	0	38,547	45,700	45.8%
1/2	1/2	0	0	38,738	45,509	46.0%
1/3	2/3	0	0	38,780	45,467	46.0%
0	0	1	0	40,948	43,299	48.6%
1/17	0	16/17	0	41,335	42,912	49.1%
1/2	0	1/2	0	41,920	42,327	49.8%
1/3	0	2/3	0	41,963	42,284	49.8%
0	0	0	1	42,574	41,673	50.5%
0	0	1/17	16/17	42,846	41,401	50.9%
0	0	1/2	1/2	43,397	40,850	51.5%
1/19	1/19	1/19	16/19	43,491	40,756	51.6%
1/4	1/4	1/4	1/4	44,439	39,808	52.7%
2/7	0	1/7	4/7	44,525	39,722	52.9%
1/4	1/8	1/8	1/2	44,615	39,632	53.0%

FIGURE 6.8. Performance on the task of preposition guessing evaluated on a development set extracted from the BNC for different values of the coefficients λ_i. 'Correct' means that the preposition found to be most probable is the one that was actually used in the BNC; 'incorrect' means that it is not the one actually used in the BNC (although there are often more than one possibility, so the preposition chosen by the system is not necessarily unreasonable). Each of the coefficients λ_1, λ_2, λ_3 and λ_4 was set to ¼, ½, 1, 2 and 4 (normalised in the table above); the table shows a small number of illustrative results, sorted according to performance. The best performance, 53.0% accuracy, is obtained for a particular set of non-zero values of λ_i, compared to 52.7% for equal values of $\lambda_i = 1/4$. Without clustering, the result is 40.4% accuracy, whereas using the clusters in lieu of individual counts gives an accuracy of 50.5%.

6.5.2. Detecting preposition errors

According to IZUMI *& al.*, the preposition error rate for learners of English can be as high as 10 per cent (2003); in other words, every tenth preposition is wrong. Later publications on automatic error detection have reported performance on test sets with equally high error rates, and we chose to use a test set with 10 per cent error rate as well in order to obtain comparable results at least in that respect.

Our test set consists of about 1,500 incorrect instances extracted from the CLC, as well as nine times as many instances which are not marked up as erroneous and therefore taken to be correct. The (v, p, n) triples are all extracted from phrases where the preposition immediately follows the verb to minimise the possibility of syntactic misanalysis.

It is important to realise that preposition guessing and error detection are two distinct problems, and that a system with good performance on the former is not on its own sufficient for the latter task. For the sake of example, let us assume that we have at our disposal a system which is able to predict the correct preposition 70 per cent of the time, and that we want to detect incorrect prepositions in a text where 10 per cent of the prepositions are wrong. One might consider the original preposition to be incorrect when the system chooses a different

preposition, which would give the following confusion matrix (assuming that there is only one correct preposition in a given context):

	Correct	Incorrect	Σ
Classified 'correct'	0.63	$0.03 - \varepsilon$	$0.66 - \varepsilon$
Classified 'incorrect'	0.27	$0.07 + \varepsilon$	$0.34 + \varepsilon$
Σ	0.90	0.10	1

The proportion ε represents the cases where the system has predicted an incorrect preposition distinct from the original one. As can be seen, the over-all precision may be as low as 0.07/0.34 or 21 per cent, which means that the system will give four false positives for each incorrect preposition detected, making it practically unusable. In general, precision is considered to be more important than recall for this task.

One way of achieving higher precision is to consider a preposition q as incorrect only in the case of there being ample evidence for another preposition in the same context, for instance if there exists a preposition p^* such that

$$\frac{\nu(v, p^*, n)}{\sum_p \nu(v, p, n) + 1} > \frac{\nu(v, q, n) + 1}{\sum_p \nu(v, p, n) + 1} + \pi_0,$$

where one occurrence of the instance under consideration (v, q, n) has been added to the BNC-based counts ν in order to deal more gracefully with instances not observed in the BNC. To avoid confusion with previous notation, $\bar{\pi}$ will be used for thus modified probabilities based on BNC counts, which enables us to rewrite the inequality as

$$\bar{\pi}(p^*|v, n) > \bar{\pi}(q|v, n) + \pi_0.$$

The dotted line in Fig. 6.9 shows precision and recall for this method.

Both approaches for making use of the clusters as described earlier in the case of preposition guessing (*viz,* fall-back and smoothing) can be adapted to work for error detection as well. In the first case, q will be considered incorrect if at least one of the following relations holds:

$$\bar{\pi}(p^*|v, n) > \bar{\pi}(q|v, n) + \pi_1$$

$$\bar{\pi}(p^*|V, n) > \bar{\pi}(q|V, n) + \pi_2$$

$$\bar{\pi}(p^*|v, N) > \bar{\pi}(q|v, N) + \pi_3$$

$$\bar{\pi}(p^*|V, N) > \bar{\pi}(q|V, N) + \pi_4$$

In the second case, q will be considered incorrect if

$$\bar{\pi}(p^*|v \in V, n \in N) > \bar{\pi}(q|v \in V, n \in N) + \pi_0,$$

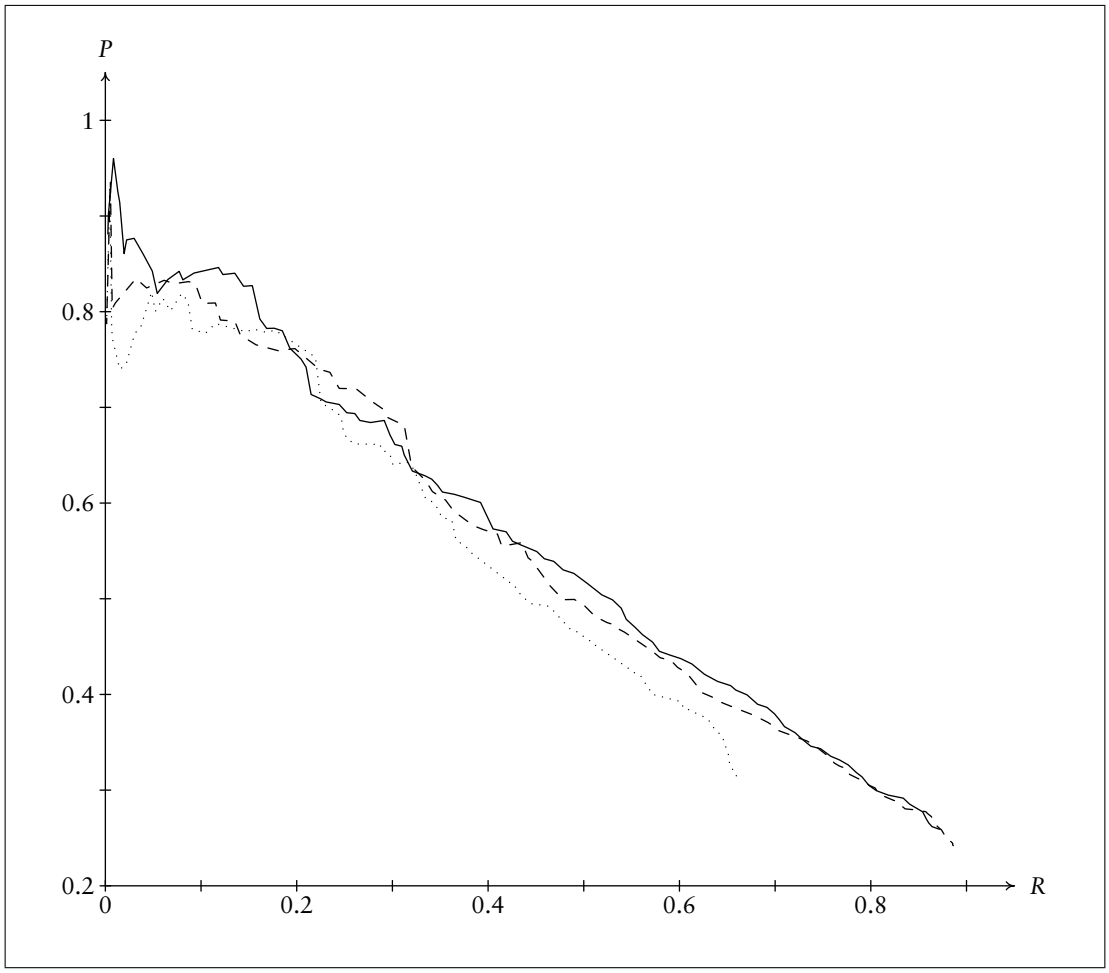

FIGURE 6.9. Precision–recall curve for error detection without clustering (dotted), with cluster fall-back (dashed), and with cluster smoothing (continuous line).

where

$$\bar{\pi}(p|v \in V, n \in N) = \lambda_1 \bar{\pi}(p|v,n) + \lambda_2 \bar{\pi}(p|V,n) + \lambda_3 \bar{\pi}(p|v,N) + \lambda_4 \bar{\pi}(p|V,N).$$

Fig. 6.9 shows the results for both methods with $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/4$ and for different values of the parameter $\pi_1 = \pi_2 = \pi_3 = \pi_4$ in the first case and π_0 in the second case.

Both methods give a noticeable increase in performance, roughly a five percentage points' increase in precision for a given recall value over large parts of the precision–recall curve, though admittedly neither method consistently outperforms the baseline (no clusters) for all recall points. This shows that verb/noun clustering can provide increased performance in the task of detecting prepositional choice errors, and we hope that further enhancements, such as training the parameters λ_i and π_i or finding a way to combine the two methods, can provide further improvements.

6.6. Discussion and future work

The smoothing method in particular allows a significant increase in recall at high precision; with 15.4 per cent recall at 82.7 per cent precision, the numbers actually outperform the state of the art as exemplified by TETREAULT & CHODOROW's 14.1 per cent recall at 82.1 per cent precision (2008). These results are based on different test sets and thus not directly comparable, but it is still interesting to have a look at methodological differences. TETREAULT & CHODOROW's test set presumably contains all kinds of prepositional choice errors, whereas ours is restricted to prepositions with a verbal head and a nominal dependent, a subset for which the choice of preposition can be expected to be more constrained, but which also excludes a number of errors that are both frequent and trivial to detect (*e.g.,* the adverbial **on/at Tuesday*), thus potentially contributing to high precision and reasonable recall. As features, they used the head of the preceding verb phrase and following noun phrase, approximating the verb and noun features used in our model, but also added two words on either side of the preposition and their parts of speech as features, along with a score of others; furthermore, heuristic rules were added, such as the one excluding antonyms as potential corrections, or the one preventing the system from ever considering '*for* + human noun' as incorrect given that it can allegedly occur quite freely in contexts where a different preposition might be expected, but would change the meaning completely; finally, the problem of detecting extraneous prepositions was added in, which seems likely to have caused an increase in performance given a large proportion of trivial errors like *many *of people* and *friends with *with*, though this is difficult to ascertain from the paper. To give two concrete examples from the CLC that may be representative of frequent trivial errors in other learner corpora as well, consider that as many as 1.2 per cent of the extraneous prepositions in single-error sentences occur as part of the phrase *to (tele)phone *to the,* and that over 0.7 per cent of the prepositional choice errors involve the expression *to look forward to*. Additional features and heuristic rules would seem to be complementary to clustering techniques such as the ones presented in this paper, and combining the two is an area for further research.

As for the clustering itself, using the prepositions as features for clustering verbs and nouns independently has been shown to improve the performance in the tasks of preposition correction and guessing, but there is clearly a potential for better clustering based on additional features. Verbs and nouns interact via the preposition, which is not captured by the two independent clustering models used in our work. It would be possible to supplement the prepositional features used for verb clustering with the nouns and *vice versa,* but such a process would result in an extremely sparse feature set. Clustering the verbs first and using the verb clusters instead to refine the prepositional features used for noun clustering would cause asymmetry between the noun and verb clustering, and the same issue would of course apply if it were done the other way round. A more elegant solution would be a bi-clustering approach, such as the infinite relational model (KEMP *& al.* 2006).

Chapter 7.

Semi-automatic annotation

Manual error annotation of learner corpora is time-consuming and error-prone, whereas existing automatic techniques cannot reliably detect and correct all types of error. In this chapter, we shall investigate the feasibility of integrating automatic methods such as the ones described in previous chapters into the annotation process and thereby obtaining annotation of higher quality with less manual effort.

7.1. Status quo

The number of sizeable error-annotated corpora remains limited, at least partly because 'error annotation is one of the most tedious, subjective, time-consuming and labo[u]r-intensive aspects of corpus annotation' (Wible *& al.* 2001). The clc is, with over 16½ million words error-coded, one of the largest corpora of this kind currently in existence. The initial error annotation is done manually by a small team of trained annotators who type sgml tags and proposed corrections using a standard text editor, followed by a post-editing step designed to detect not only occasional sgml errors, but also inconsistencies in the annotation (referred to as 'detoxification' by the error coders). A fairly pragmatic and linguistically superficial set of error tags 'designed in such a way as to overcome [...] problems with the indeterminacy of some error types' (Nicholls 2003) and a reasonably thorough coding manual further contribute to increasing the annotation quality. The error annotation was originally performed by one person; with the expansion of the team, this person no longer does any of the initial annotation, but instead reads through all the annotated scripts to assure a high level of consistency. Nevertheless, we are not aware of any formal inter-annotator agreement studies having been performed, and it is of course difficult to obtain

a quantitative measure of how good the annotation actually is. It should also be noted that certain errors are not marked up as such in the CLC, in particular misspelt proper names (apart from the most well-known ones, *e.g.*, **Ingland* and **Amrica*) and errors which appear to be directly induced by the exam question; this is a sensible compromise seen from a textbook writer's perspective, but it inevitably causes certain classes of incontestable error not to be marked up and thus to remain indistinguishable from correct text, which makes the corpus less suitable for other applications. Certain other practices are suboptimal because they make it difficult to tell what exactly was part of the original examination script and what has been modified later, for instance the somewhat crude anonymisation technique that consists in substituting a number of *x*'s for proper names and other potentially sensitive information; given the types of information that is typically removed by this means, simply considering *x*'s as a noun phrase often works in practice, though.

To get an idea of the level of consistency in some fairly clear-cut cases, we identified words and phrases often marked up as erroneous either unconditionally or in given linear contexts and checked whether remaining identical occurrences actually should have been marked up as well. As Fig. 7.1 shows, the level of consistency is generally high for obvious errors. However, a trivial error like *occured* spelt with one *r* has actually been missed 15 per cent of the times it occurs, which seems to indicate that a hybrid system where simple errors are marked up automatically could usefully complement the human annotator by spotting, in particular, typographical errors and others that are easily overlooked. Furthermore, it turns out that certain trivial errors are inordinately frequent (*cannot* incorrectly split into *can not* alone accounts for 0.2 per cent of the errors), which implies that even a relatively crude system would be able to deal with a meaningful subset of the errors and let the human annotator concentrate on more interesting/complex ones. As for more subtle details, upon which style guides are likely to disagree, the lower consistency rates arguably indicate that the corresponding putative rules are not universally followed; whether or not *Third World* should be capitalised is purely conventional, and there is no obvious reason for requiring a comma in *on the other hand, I want to improve my conditions of employment,* but not in *on the other hand I agree with the complaints,* though it is difficult to tell from the corpus whether the annotators have attempted always to require a comma after sentence-initial adverbials and sometimes forgotten, or only require it when its absence would cause ambiguity, at least locally ('garden paths'), and sometimes added commas which are not strictly necessary. Another possible explanation could be that the annotators try to render the use of commas consistent within each script rather than globally, though such an approach seems unlikely to result in high correction rates for specific expressions; or the policy could have changed as the consequence of a shift in usage observed amongst professional writers and publishers (this was reportedly the case for hyphens in attributive compounds, which are no longer considered compulsory in general). Whatever the cause might be, inconsistencies make the corpus less suitable as training data or as a gold standard of errors for an automatic system to detect; such issues are to a certain extent inherent in the

Original	Correction	Rate
accomodation	accommodation	99%
a lots of	a lot of	99%
forward to hear	forward to hearing	99%
Your faithfully	Yours faithfully	>96%
appreciate if	appreciate it if	96%
to spent	to spend	95%
center	centre	91%
However there	However, there	87%
a part time	a part-time	86%
occured	occurred	85%
On one hand	On the one hand	60%
third world	Third World	57%
other hand I	other hand, I	~50%

FIGURE 7.1. Correction rate, the proportion of incorrect occurrences of a word/phrase that are actually marked up. (Some of the words/phrases that should typically be corrected may be correct in specific cases; such instances were excluded before the rates were calculated.)

task of error annotation, but it is to be hoped that more sophisticated consistency checks can contribute to detect current inconsistencies, leading the way to clearer guidelines, at least some of which can be enforced mechanically.

7.2. Automatic pre-annotation

Despite considerable work on methods and systems to detect and correct spelling and grammar errors, none of the existing error-annotated corpora seem to have been prepared using such techniques. In order to investigate the potential of semi-automatic annotation in terms of making the human annotator's task less laborious and repetitive, a system was developed that aims to detect trivial or frequent errors automatically and add the corresponding annotation, including corrections when appropriate (this section) and combined with an error annotation tool to make the error annotation more readable for the annotator and easier to manipulate (next section).

7.2.1. Purveyors of perpetual perplexity

Many trivial errors are committed — and corrected — over and over again, such as the ones shown in Fig. 7.1. For the purposes of this experiment, rules for automatic correction of common errors were derived directly from the existing error annotation: a correction rule was created for errors that appear at least 5 times and are corrected in the same way at least 90 per cent of the time. In addition to the text marked up as erroneous, up to one word on either side was used to model the immediate context in which an error occurs. For instance, the correction *I* <sx>*thing*|*think*</sx> *that* would give rise to four potential indicators of

error, *thing, I thing, thing that* and *I thing that,* each of which can be searched for and counted in the corrected text; in this case, the result of such an investigation would be that at least *I thing that* is non-existent or extremely rare in the corrected text and thus a good indicator of error, and furthermore that the error is always or most of the time corrected in the same way (*i.e.*, to *I think that*); the conclusion would be that an automatic system ought to hypothesise every occurrence of *I thing that* as a misspelling of *I think that*.

The following examples illustrate the kinds of error that can be detected and corrected using such simple rules; the previously discussed example appears as *I* <SX>*thing|think*</SX> *that,* which means that any occurrence of *I thing that* will result in *thing* being marked up as a spelling confusion (SX) error for *think*.

No context:
<S>accomodation|accommodation</S>
<SA>center|centre</SA>
<RP>french|French</RP>
<RP>an other|another</RP>
<UP>I'am|I am</UP>
<MP>above mentioned|above-mentioned</MP>
<W>be also|also be</W>
<ID>In the other hand|On the other hand</ID>

Left context:
the <RP>internet|Internet</RP>
reason <RT>of|for</RT>
all <AGN>kind|kinds</AGN>
I <SX>though|thought</SX>
despite <UT>of|</UT>
computer <RN>programme|program</RN>
to <DV>complaint|complain</DV>

Right context:
<DA>Your|Yours</DA> Sincerely
<AGD>this|these</AGD> things
<MP>long distance|long-distance</MP> travel
<UV>be|</UV> appreciate
<RV>loose|lose</RV> their

Left and right context:
50 <MP>years|years'</MP> experience
I <SX>thing|think</SX> that
I <DV>advice|advise</DV> you
a <DJ>slightly|slight</DJ> increase
is <SX>to|too</SX> small

As we have seen, certain errors are not always corrected even if they should have been, so requiring 100 per cent correction rates would hardly induce any rules at all. Furthermore, a rule can be useful even if it occasionally 'corrects' what was correct in the first place since it is less arduous for the human annotator to remove incorrect error mark-up from time

to time than to add it in the common case (*e.g.,* a rule that indiscriminatively annotates *can not* as a misspelling of *cannot* will fail in cases like *can not only ..., but also,* but this is of little consequence since incorrect instances of *can not* in the CLC outnumber correct ones by almost two orders of magnitude). One should keep in mind, though, that human annotators reportedly find spurious errors introduced automatically particularly annoying, so an imperfect rule should only be added when the resulting annotation is correct in an overwhelming majority of the cases; actually, 90 per cent is probably far too low from that perspective, but the previous considerations of annotation consistency make it seem likely that errors which have been corrected by the annotators 90 per cent of the time really should be corrected more often, so we expect the rules to be somewhat more reliable than the threshold suggests. Unfortunately, the threshold chosen precludes some obvious errors from being identified (*e.g., *occured*), but then a lower threshold could easily lead to too many spurious errors for the human annotator to remove. (Some of these errors will in any case be identified by other methods, as described in the following sections.) Manual evaluation of specific rules might be worthwhile if such a system is to be employed on a large-scale annotation project, but would clearly require a fair amount of work by someone who can make policy decisions on what should and should not be marked up as erroneous and was not feasible within the scope of this study.

7.2.2. Morphological metamorphosis

The corpus-derived rules described in the previous section work well for specific words which are both frequent and frequently misspelt in the CLC, but do not generalise to similar or even virtually identical errors involving different words. Travel and tourism seem to be a popular topic in Cambridge exams, so the misspelling of *travelled* as **traveled* with one *l* is amply exemplified, whereas **signaled* occur only once, so no corresponding correction rule will be generated when using the proposed method and thresholds, and no rule can be derived from the corpus for a word like **groveled,* which does not occur at all, but might well appear in the future; these errors all have to do with British English rules for l-doubling in morphological derivatives, and they can therefore be handled systematically, provided we have access to a word's correct morphology.

Without trying to make the corrected exam scripts conform in all respects to CUP's house style, the CLC annotators naturally use Cambridge dictionaries to settle any doubts regarding orthography and morphology, albeit reluctantly in cases where the last editions do not yet reflect what is about to become established usage. It would therefore be preferable to use a Cambridge dictionary as the basis for automatic annotation rules; unfortunately, though, the ones available to us do not contain sufficient information on inflectional morphology, so we had to use a different data source and chose the Lexical Database developed by the Dutch Centre for Lexical Information (CELEX), which in addition contains useful information about noun countability and derivational morphology.

The examples below illustrate the types of error that can be automatically detected and corrected by predicting systematic morphological anomalies modelled on actual errors found in the CLC.

Plural of singulare tantum:
<CN>abhorrences|abhorrence</CN>
<CN>bigamies|bigamy</CN>
<CN>blamelessnesses|blamelessness</CN>

Derivation of adjective:
<DJ>academical|academic</DJ>
<DJ>atypic|atypical</DJ>
<DJ>cheerfull|cheerful</DJ>
<DJ>non-legal|illegal</DJ>
<DJ>inlegible|illegible</DJ>
<DJ>unmature|immature</DJ>
<DJ>inpossible|impossible</DJ>
<DJ>inrational|irrational</DJ>
<DJ>uncommissioned|non-commissioned</DJ>
<DJ>incertain|uncertain</DJ>

Derivation of adverb:
<DY>abnormaly|abnormally</DY>
<DY>academicly|academically</DY>
<DY>accidently|accidentally</DY>
<DY>accuratly|accurately</DY>
<DY>angryly|angrily</DY>
<DY>barily|barely<DY>
<DY>closelly|closely</DY>
<DY>wishfuly|wishfully</DY>

Adjective inflection:
<IJ>biger|bigger</IJ>
<IJ>brainyer|brainier</IJ>
<IJ>crazyest|craziest</IJ>
<IJ>grimest|grimmest</IJ>
<IJ>Chineses|Chinese</IJ>

Noun inflection:
<IN>addendas|addenda</IN>
<IN>addendums|addenda</IN>
<IN>alumnas|alumnæ</IN>
<IN>amanuensises|amanuenses</IN>
<IN>anthologys|anthologies</IN>
<IN>antiheros|antiheroes</IN>
<IN>bagsfuls|bagsful</IN>
<IN>boleroes|boleros</IN>
<IN>nucleuses|nuclei</IN>
<IN>oxes|oxen</IN>
<IN>paterfamiliases|patresfamilias</IN>
<IN>persona non gratas|personæ non gratæ</IN>
<IN>schemas|schemata</IN>

<IN>tooths|teeth</IN>
<IN>aircrafts|aircraft</IN>

Verb inflection:
<IV>abandonning|abandoning</IV>
<IV>abbreviateing|abbreviating</IV>
<IV>abhoring|abhorring</IV>
<IV>abolishs|abolishes</IV>
<IV>accompanys|accompanies</IV>
<IV>amplifis|amplifies</IV>
<IV>abolishd|abolished</IV>
<IV>abolisht|abolished</IV>
<IV>abstainned|abstained</IV>
<IV>accompanyed|accompanied</IV>
<IV>ferrid|ferried</IV>
<IV>airdroped|airdropped</IV>
<IV>breeded|bred</IV>
<IV>slidden|slid</IV>

7.2.3. Spell-catching

The CELEX database distinguishes between British and American spellings, so a list of American words which do not exist in British English can be derived as well:

<SA>britches|breeches</SA>
<SA>jewelry|jewellery</SA>
<SA>maneuver|manœuvre</SA>
<SA>Cesarean|Cæsarean</SA>

Proper names and other words always written with a capital letter were extracted from the database to deal with capitalisation errors:

<RP>gouda|Gouda</RP>
<RP>teutonic|Teutonic</RP>
<RP>euclid|Euclid</RP>
<RP>scotland|Scotland</RP>
<RP>christmastime|Christmastime</RP>

Finally, a list of correct word forms was extracted from the database to enable detection of mundane spelling errors: words consisting entirely of lowercase letters not already corrected in one of the previous steps and not in the wordlist can be identified as likely typographical errors. For such words, the correct spelling is unknown, and a distinguished token *(¿?)* takes the place of a correction to indicate this. It would of course be possible to make the system propose a plausible correction, for instance by using methods like the ones proposed by DEOROWICZ & CIURA to model the kinds of errors typically committed (2005),

relying on statistics from the CLC for error frequencies; this would clearly be useful in a tool aimed at less confident language users and would be an interesting extension to the system, but seems less important in the context of error annotation and is unlikely to have a significant impact on the annotation speed given that all frequent errors with obvious corrections will have been handled by the corpus-derived rules. Misspelt words containing at least one capital letter are not detected, partly because we have not tried to compile a comprehensive lexicon of names, partly because of the CLC policy of not correcting proper names in general.

7.2.4. Euphonia

We have previously (*see* Sect. 4.4) obtained good results with machine-learning techniques on the task of detecting incorrect choice between the two euphonic variants of the indefinite article *(a/an)*. The rule is to use *a* before a consonant and *an* before a vowel; the CELEX database provides pronunciations and we used the database for other error types already, so it seemed natural to take advantage of that information. Only the first sound in a word is significant for the choice of *a* or *an,* and only whether it is consonantal or vocalic, as illustrated in the following examples:

minister consonant
MP vowel
open vowel
one consonant
home consonant
hour vowel
hotel vowel/consonant
utter vowel
useful consonant
Uruguay vowel/consonant

Words with alternative pronunciations, such as *Uruguay,* may be used with either form of the article. (The traditional usage of *an* in front of unaccented aspirated *h* is not accounted for, but the information needed is available in the CELEX database, so this could easily be added.) The text is part-of-speech-tagged with RASP (BRISCOE, CARROLL & WATSON 2006) before this step to avoid conflation of unrelated occurrences of *a* with the indefinite article.

7.2.5. Synopsis

The flowchart in Fig. 7.2 illustrates how the different parts of the system interact to produce automatic annotation. The contents of all the intermediate files produced from a short example file can be found in Appendix D.3. Each exam script in the CLC contains information about the candidate and the exam taken, as well as the actual text written:

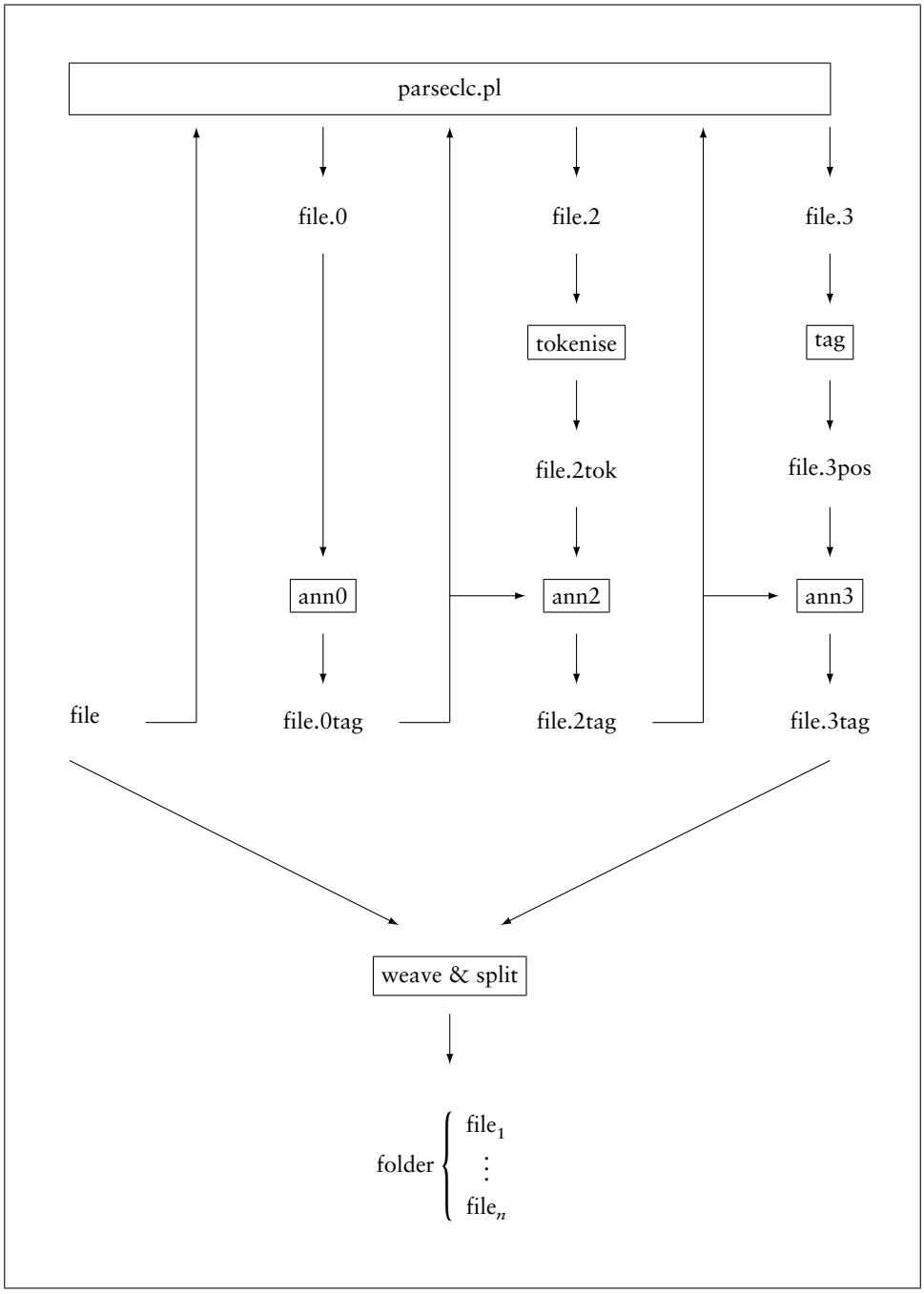

FIGURE 7.2. Schematic overview of the automatic annotation process, starting with a single file containing multiple unannotated exam scripts and ending up with a set of files, each containing an annotated script.

<candidate>*[metadata about examination and examinee]*</candidate>
<text>
<answer1><question_number>9</question_number> *[...]* <original_answer>
I wont be going to the little nice persian café this after noon becauce I eated to much for lunch and have now a awfull stomachache.
</original_answer></answer1>
[more answers]
</text>

parseclc extracts the text to analyse:

I wont be going to the little nice persian café this after noon becauce I eated to much for lunch and have now a awfull stomachache.

— file.0

The only detail worth mentioning at this point is the normalisation of *é* to *é;* XML provides many ways to represent a given character, and ensuring that *é* never appears as for instance *é* or *é* simplifies further processing. The first error tags are then added by *ann0*, using simple string matching:

I <e t="MP">wont<c>won't</c></e> be going to the
<e t="W">little nice<c>nice little</c></e> persian café this
<e t="RP">after noon<c>afternoon</c></e>
<e t="S">becauce<c>because</c></e> I eated
<e t="SX">to<c>too</c></e> much for lunch and
<e t="W">have now<c>now have</c></e> a
<e t="DJ">awfull<c>awful</c></e>
<e t="RP">stomachache<c>stomach ache</c></e>.

— file.0tag

At this point, error tags have been added, which will have to be removed before the next processing step. *parseclc* again extracts the text, using the corrections rather than the original text when applicable:

I won't be going to the nice little persian café this afternoon because I eated too much for lunch and now have a awful stomach ache.

— file.2

This partly corrected version of the text is then passed through RASP's sentence splitter and tokeniser. *ann2* detects further errors based on the tokens and adds annotation. The process is repeated once more, this time adding part-of-speech tags for *ann3* to work on, which gives the following annotated output:

I <e t="MP">wont<c>won't</c></e> be going to the
<e t="W">little nice<c>nice little</c></e>

```
<e t="RP"><i>persian</i><c>Persian</c></e> café this
<e t="RP"><i>after noon</i><c>afternoon</c></e>
<e t="S"><i>becauce</i><c>because</c></e> I
<e t="IV"><i>eated</i><c>ate</c></e>
<e t="SX"><i>to</i><c>too</c></e> much for lunch and
<e t="W"><i>have now</i><c>now have</c></e>
<e t="FD"><i>a</i><c>an</c></e> <e t="DJ"><i>awfull</i><c>awful</c></e>
<e t="RP"><i>stomachache</i><c>stomach ache</c></e>.</p>
```

— file.3tag

Because the XML mark-up is handled properly, the fact that *awful* is embedded within an error tag does not prevent the system from detecting that the preceding determiner should be *an* rather than *a*. If *parseclc* were applied again, the following output would have been generated:

```
<p>I won't be going to the nice little Persian café this afternoon because I ate too much for lunch
and now have an awful stomach ache.</p>
```

The process can obviously continue with, for instance, the generation of syntactic annotation as the basis of additional error annotators, such as the ones discussed in the previous chapter. For the purposes of this experiment, though, the output from *ann3* was combined with the original file (containing metadata irrelevant for the automatic error annotation) to create complete automatically annotated files for the human annotator to work on.

7.3. Annotation tool

Whereas some corpora have been annotated using dedicated tools such as the Université Catholique de Louvain Error Editor (UCLEE, *cf.* DAGNEAUX, DENNESS & GRANGER 1998), the CLC annotators have written SGML tags directly in a text editor. This is not necessarily an impediment to efficient annotation when compared to systems which require error tags to be selected from menus and submenus given that the coding scheme uses short codes and makes judicious use of SGML abbreviation techniques in order to limit the number of characters and thus keystrokes needed to mark up an error. The code is also quite readable as long as there are not too many nested errors, but occasional SGML errors, which render the entire file in which they occur unparseable until the error has been corrected, are nevertheless difficult to avoid completely.

An additional consideration for semi-automatic annotation is the ease with which an incorrect error tag added by the machine can be removed by the human.

We felt that a simple annotation tool was the right solution to these problems: it would provide a graphical representation of the error annotation, making it easier for the annotator to see where each error begins and ends, in particular in the case of nested errors; the

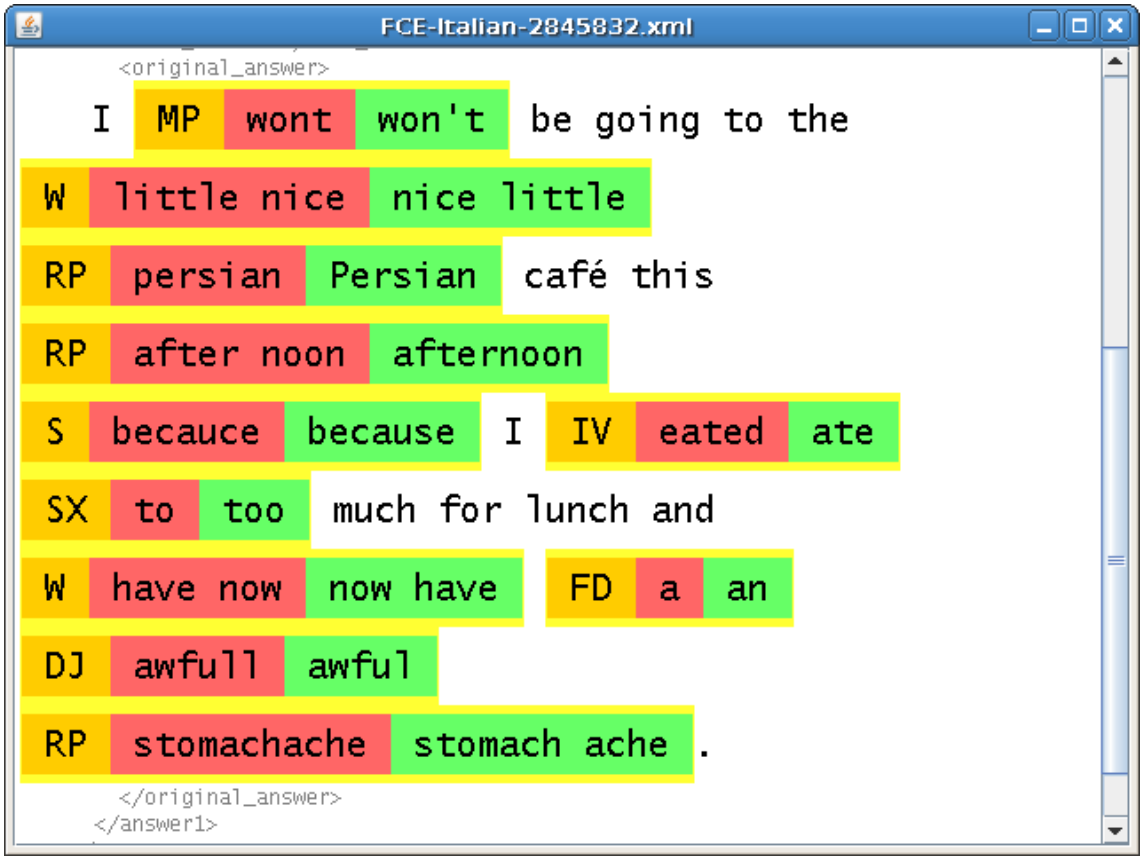

FIGURE 7.3. The pre-annotated example sentence as it appears in the annotation tool. For each error annotation, the error type is shown to the left, on an orange background; the error in the middle, on a red background; and the correction to the right, on a green background.

number of keystrokes needed could be reduced further, and the need for typing 'exotic' characters eliminated; SGML errors would never appear; and one keystroke would be sufficient to remove an unwanted error tag.

Fig. 7.3 shows how a sentence with error annotations appears in the annotation tool.

7.4. Annotation experiment

The head annotator of the CLC kindly agreed to annotate text taken from previously unannotated parts of the corpus using the system described on the previous pages. After initial testing and development, four different set-ups were tried, as described in the following sections, in order to investigate the contribution of different factors.

	Words	Tags	Words/tag	Hours	Words/hour	Tags/hour
CLC coding	6,736,452	746,252	9	5,156	1,306	145
—— & detox	——	——	——	6,924	972	108
Part 1	13,127	934	14	4	3,281	233
Part 2	19,716	1,433	13.8	?4	4,929	358
Part 3A	9,881	311	31.7	1.51	6,544	206
Part 3B	9,679	1,023	9.46	2.71	3,572	377
Part 3 (Σ)	19,560	1,355	14.65	4.22	4,635	316
Part 4	18,610	1,373	13.55	1.66	11,210	827

FIGURE 7.4. Performance in terms of annotation speed. The first two lines of the table relate to the part of the CLC that has been error-coded during the last couple of years, the figures on the first line only including the time spent on the initial coding, the second line including the subsequent post-editing step ('detoxification' to remove SGML errors and coding inconsistencies) as well; the remaining lines relate to the annotation produced as part of the annotation experiment described in this chapter, parts 1–4 being described in Sect. 7.4.1–7.4.4. The number of words and tags is indicated for each part of the corpus, and the inverse tag density (words per tag) is calculated to give an idea of the amount of errors (more words per tag means fewer errors, higher-quality text and less work for the annotator). The number of hours spent to annotate (including post-editing in the case of the second line) each part is indicated, which, in combination with word and tag counts mentioned previously, allow the annotation speed to be calculated in terms of wordsper hour as well as tags per hour.

7.4.1. Manual annotation (part 1)

Statistics from previous years of CLC annotation enable us to estimate average annotation speed in terms of tags per hour or words per hour. We were concerned that those data points might not be directly comparable with the ones obtained as part of the experiment, though, and therefore included a batch of scripts for manual annotation, asking the annotator to type tags in a text editor as previously.

As expected, a few SGML errors appeared:

> Mismatched tags:
> <#DK>competitable|competitive</#DJ>
> <#RJ>fashion|fashionable</#DJ>
> <#SA>humor|humour</#SX>
> <#FV>making|to make</#RV>
>
> Missing angle bracket:
> <#DJ>successfull|successful</#DJ
>
> Complex error:
> <#UV>I'm</#I> (for <#UV>Im|I</#UV>)

There were also some overlooked[29] errors which the automatic system would have detected:

> <RP>clare|Clare</RP>
> <RP>10'000|10,000</RP>

[29]) The annotator later told that the first two errors were deliberately ignored.

	Before	Correct	After	*P*	*R*
Part 2	372	345	1,448	93%	24%
Part 3A	0		280		
Part 3B	397	353	1,023		
Part 3 (Σ)				89%	27%
Part 4	1,302	1,293	1,373	99%	94%

FIGURE 7.5. Performance in terms of precision and recall of errors by the pre-annotation system measured against the human annotator. The *before* column indicates the number of tags added during the pre-annotation step; the *after* column indicates the total number of errors after annotation; and the *correct* column indicates the intersection between the two sets (*i.e.*, the number of tags added during pre-annotation that were not subsequently removed during annotation). Note that Part 4 uses human pre-annotation (resulting from the Part 3 annotation).

<SA>analyze|analyse</SA>
<SA>analyzed|analysed</SA>
an <MP>all time|all-time</MP> low
our <MP>day to day|day-to-day</MP> life
it is <SX>to|too</SX> complicated
had to <RV>seat|sit</RV> in the back row

As for the annotation speeds in this experiment compared to previous annotation of the CLC, the two turned out to be significantly different (*see* Fig. 7.4); this can at least in part be ascribed to better English with fewer errors in the experiment (on average 1 error tag added per 14 words) than in previously annotated parts of the corpus (1 tag per 9 words), and is thus not entirely surprising, but it also shows that any direct comparison with previous years' results is likely to be misleading.

7.4.2. Semi-automatic annotation (part 2)

For the second part of the experiment, scripts were pre-annotated automatically using the system described in Sect. 7.2 before it was given to the human annotator for correction and supplementation using the annotation tool. Examination of the final annotation showed that the automatic pre-annotation system had precision of 93 per cent and recall of 24 per cent (*see* Fig. 7.5), which is quite encouraging given that the system is neither comprehensive nor fine-tuned: increased recall without loss in precision can be obtained by extending the system's coverage, and precision is impeded by an incomplete and slightly outdated lexicon. The annotation speed turned out to be about 50 per cent higher than in the previous experiment; in addition to this, there are no SGML errors to correct, and the annotation is more consistent, which eliminates the need for subsequent SGML verification and vastly reduces the need for consistency checking, thus making the effective speed increase closer to 100 per cent.

		Tagged sentences					Classification	
	Total	Before	Correct	After	*P*	*R*	*P*	*R*
Part 3A	686	0		198		(71%)		
Part 3B	517	297	271	417	91%		81%	
Part 3 (Σ)						44%		68%

FIGURE 7.6. Performance in terms of the system's ability to detect sentences containing at least one error. The *total* column shows the total number of sentences. The *before, correct* and *after* columns have the same meaning as in Fig. 7.5, but they refer to a number of sentences rather than a number of individual errors.

7.4.3. Annotation of individual sentences classified as good/bad (part 3)

In order to get a better idea of how important context is for correct annotation, as well as to assess the potential for more efficient annotation by focusing on sentences more prone to contain errors, sentences were split into two sets, likely correct (3A) and likely containing errors (3B), for the third part of the experiment. As one would expect, this set-up caused the annotation speed in terms of words per hour to increase for the largely correct sentences, and in terms of error tags per hour for the largely incorrect sentences, whilst both performance measures declined globally, at least partly because it is more burdensome and mentally exhausting for the annotator to deal with individual sentences than connected passages of discourse. Sentence-level performance is indicated in Fig. 7.6.

7.4.4. Re-evaluation in context (part 4)

Finally, the sentences from part 3 were put together again and presented to the annotator anew for evaluation in context. This gave a precision figure for manual detection of errors in individual sentences out of context of well over 99 per cent, whereas recall was a bit lower at 94 per cent; the conclusion from this is that what appear to be errors when a sentence is regarded in isolation usually turn out to be errors in context as well, whereas about 1 out of 20 errors, including a high proportion of tense errors, require extra-sentential information to be detected. This part of the experiment also permitted us to calculate an upper bound for annotation speed given very high-quality pre-annotation: compared to part 1, there was an increase of 250 per cent in annotation speed, or approaching 325 per cent if the amount of post-editing can be reduced, although the fact that the annotator had already seen the sentences, albeit out of order, may also have contributed to the speed increase observed in this part of the experiment..

7.5. Conclusion

We have seen that an annotation tool that incorporates automatic techniques for error detection and correction can contribute to higher accuracy and increased productivity in

the task of error annotation. This is significant since manual error annotation can be both laborious and tedious, whereas the existence of sizeable error-annotated corpora is crucial both for the study of language containing errors (be it from a pedagogic or a more purely linguistic perspective) and for the development of 'grammar checkers' and other tools that actually address the areas of language that can be shown to be problematic. Higher recall can obviously be achieved by adding error detectors for further types of error, and the techniques currently employed can also be refined to give higher performance.

Chapter 8.

Conclusion

As laid out in section 1.5, the main goal of this doctoral work has been to develop methods for automatic detection and correction of grammatical errors.

Given that corpus material is used not only to validate the detection techniques, but also in many instances constitutes the evidence upon which they rely to distinguish between right and wrong, corpus encoding techniques and methods that enable us to make more effective use of corpora are essential. In Chapter 3, we presented a way to add linguistic annotation provided by a parser to a corpus in an XML format. The parsed version of the BNC thus obtained was put into use in different ways in subsequent chapters, and has also been used by other researchers for other purposes (recent papers mention work on semantic role labelling, Fürstenau & Lapata 2009, and logical metonymy, Shutova 2009).

In Chapter 4, we presented a supervised learning approach to the binary sentence classification task and obtained an accuracy of over 70 per cent using pairs of correct and incorrect sentences from the CLC as both training and test data. Features including *n*-grams of words and parts of speech as well as grammatical relations provided by the RASP system contributed to this result, and it is expected that additional features, for instance labelled grammatical relations, would permit this generic technique to perform even better. In order to distinguish more effectively between correct sentences and ones containing a single minor mistake, more specialised error detectors can be useful. Enriching the set of features with specific indicators of particular types of error was shown to give a significant increase in accuracy, and this approach is eminently extensible to further error types.

It is difficult to evaluate our system's performance against others' since there is no generally available test set. As we saw in Chapter 5, though, others have often relied on synthetic data for training, which, according to our experiments, gives inferior performance on real errors, although faux errors are of course better than none. As for the value of synthetic data as a complement to real data for training purposes, further investigations are needed. Synthetic data has occasionally been employed as test data; this approach, however, is tantamount to the introduction of an artificial task, the performance on which may not be

elucidative of the system's ability to handle 'English as she is spoke': a representative set of real errors for testing being made publicly available would be an important step towards enabling systems to be compared against each other (although part of the problem is of course how to define such a set).

For errors that consist in an inappropriately chosen adjective or preposition, the CLC is often not on its own sufficient as a source of evidence; the preferred adjective or preposition in a given context is lexically idiosyncratic, which leads to data sparsity issues. In Chapter 6, we presented encouraging results on the task of detecting adjective choice errors based on statistics from the BNC, as well as state-of-the-art results on the task of detecting certain types of preposition errors when accumulating statistics over contexts involving similar verbs/nouns. Only magnitude adjectives were included in the experiments, so further work is needed to determine whether this approach can be extended to other classes of confusable adjectives, as well as whether clustering nouns according to adjectival preferences will be useful for less common adjective–noun combinations. The method for detecting prepositional choice errors can be extended to other frames (*e.g.*, adjective–preposition–noun and verb–preposition–noun), and it would be interesting to verify that this method, aimed at finding errors that simpler lexically based systems would overlook, can usefully complement a method that finds more trivial errors.

The idea of dedicated techniques for detection of specific errors, which may be more effective than generic approaches and furthermore allow the type of error to be identified and perhaps a plausible correction to be provided, was implemented in Chapter 7 as part of a system for semi-automatic error annotation. We showed that semi-automatic annotation can be significantly more efficient than manual annotation whilst reducing the tedium of marking up trivial errors over and over again and resulting in more consistent annotation. Handling additional error types and eliminating frequent false positives would give a system that could potentially allow an error-annotated corpus to grow faster and reach higher annotation consistency without additional resources.

Appendix A.

Error taxonomies

A.1. Gooficon classification

1. Clause skeleton
    a) Missing parts
    b) Misordered parts
2. Auxiliary system
    a) *Do*
    b) *Have* and *be*
    c) Modals
    d) Mismatch in tag questions
3. Passive
    a) Formation
    b) Use
4. Temporal conjunctions
    a) Misplacement
    b) Following clause type
    c) Predicate type
    d) Superficial tense agreement
5. Sentential complements
    a) Misordering in subordinates
    b) Subject extraposition
    c) Infinitives and gerunds
    d) Verb complement types
6. Psychological predicates
    a) Subject/object misordering
    b) Complements of reverse verbs

c) Straightforward adjectives
d) Reverse adjectives

(Source: Burt & Kiparsky 1972.)

A.2. FreeText error taxonomy

F	Form	AGL	Agglutination
		MAJ	Upper/lower case
		DIA	Diacritics
		HOM	Homonymy
		GRA	Other spelling errors
M	Morphology	MDP	Derivation-prefixation
		MDS	Derivation-suffixation
		MFL	Inflection
		MFC	Inflection-confusion
		MCO	Compounding
G	Grammar	CLA	Class
		AUX	Auxiliary
		GEN	Gender
		MOD	Mode
		NBR	Number
		PER	Person
		TPS	Tense
		VOI	Voice
		EUF	Euphony
L	Lexis	SIG	Meaning
		CPA	Adjective complementation
		CPD	Adverb complementation
		CPV	Verb complementation
		CPN	Noun complementation
		FIG	Prefab
X	Syntax	ORD	Word order
		MAN	Word missing
		RED	Word redundant
		COH	Cohesion
R	Register	RLE	Lexis
		RSY	Syntax
Y	Style	CLR	Unclear
		LOU	Heavy
Q	Punctuation	CON	Punctuation confusion
		TRO	Punctuation redundant
		OUB	Punctuation missing
Z	Typo		

The error categories above are combined with a part-of-speech tag to form a complete error description.

(Source: Granger 2003.)

A.3. SST corpus taxonomy

N	Noun	INF	Inflection
		NUM	Number
		CS	Case
		CNT	Countability
		CMP	Complement (of noun)
		LXS	Lexis
V	Verb	INF	Inflection
		AGR	Subject–verb disagreement
		FML	Form
		TNS	Tense
		ASP	Aspect
		VO	Voice
		FIN	Usage of finite/infinite verb
		NG	Negation
		QST	Question
		CMP	Complement (of verb)
		LXS	Lexis
MO	Modal Verb	LXS	Lexis
AJ	Adjective	INF	Inflection
		US	Usage of positive/comparative/superlative
		NUM	Number
		AGR	Number disagreement
		QNT	Quantitative adjective
		CMP	Complement (of adjective)
		LXS	Lexis
AV	Adverb	INF	Inflection
		US	Usage of positive/comparative/superlative
		LXS	Lexis
PRP	Preposition	CMP	Complement (of preposition)
		LXC1	Normal preposition
		LXC2	Dependent preposition
AT	Article	AT	Article
PN	Pronoun	INF	Inflection
		AGR	Number/gender disagreement
		CS	Case
		LXS	Lexis
CON	Conjunction	LXS	Lexis
REL	Relative pronoun	CS	Case
		LXS	Lexis
ITR	Interrogative	LXS	Lexis
O	Others	JE	Japanese English
		LXS	Collocation
		ODR	Misordering of words
		UK	Unknown type errors
		UIT	Unintelligible utterance

(Source: Izumi, Uchimoto & Isahara 2004.)

A.4. ICLE/Louvain taxonomy of errors

F	Form	M	Morphology
		S	Spelling
		PW	Punctuation wrong
		PM	Punctuation missing
G	Grammar	A	Articles
		ADJCS	Adjectives, comparative/superlative
		ADJN	Adjectives, number
		ADJO	Adjectives, order
		ADVO	Adverbs, order
		NC	Nouns, case
		NN	Nouns, number
		P	Pronouns
		VAUX	Verbs, auxiliaries
		VM	Verbs, morphology
		VN	Verbs, number
		VNF	Verbs, non-finite/finite
		VT	Verbs, tense
		VV	Verbs, voice
		WC	Word class
L	Lexis	CC	Conjunctions, coordinating
		CLC	Connectors, logical, complex
		CLS	Connectors, logical, single
		LC	Conjunctions, subordinating
		P	Phrase
		S	Single
		SF	Single, false friends
R	Register		[No subcategories]
S	Style		[Other]
		I	Incomplete
		U	Unclear
W	Word	M	Missing
		O	Order
		R	Redundant
X	Lexico-Grammar	ADJCO	Adjectives, complementation
		ADJPR	Adjectives, dependent preposition
		CONJCO	Conjunction, complementation
		NCO	Noun, complementation
		NPR	Noun, dependent preposition
		NUC	Noun, countable/uncountable
		PRCO	Preposition, complementation
		VCO	Verb, complementation
		VPR	Verb, dependent preposition

(Source: MacDonald Lightbound 2005.)

Appendix B.

Encoding systems

This appendix gives an overview of how characters as well as higher-level concepts such as italics, titles and sentence boundaries have been encoded in different corpora: *Brown* gives the encoding used in the Brown Corpus (Francis & Kučera 1964); *LOB* the one used in the Lancaster–Oslo/Bergen Corpus (Johansson, Leech & Goodluck 1978); *SGML* the SGML entities used in the first edition of the British National Corpus (Burnard 1995); and *XML/Unicode* the actual character or XML representation used in the last edition of the British National Corpus (Burnard 2007).

The columns *Brown*, *LOB* and *SGML* only show the characters that actually occur in the respective corpora. For accented letters, *l* represents any letter, and the *XML/Unicode* column illustrates the letter–accent combinations that occur in the BNC. The *XML/Unicode* column occasionally differ from the BNC XML edition by not replicating manifest errors. The system used for transcribing Cyrillic letters in the LOB corpus is not shown below. There is much more mark-up above the character level than what is shown below.

Brown	**LOB**	**SGML**	**XML/Unicode**
1, 2, 3. . . .	1, 2, 3, . . .	1, 2, 3, . . .	1, 2, 3, . . .
/1209	*=1209	MCCIX	MCCIX
/1209	**=1209	mccix	mccix
	1/2	&frac12	½
		&half	½
		&frac13	⅓
		&frac14	¼
		&frac15	⅕
		&frac16	⅙
		&frac17	⅐
		&frac18	⅛
		&frac19	⅑
		&frac23	⅔
		&frac25	⅖
		&frac34	¾
		&frac35	⅗
		&frac38	⅜
		&frac45	⅘
		&frac47	⅘
		&frac56	⅚
		&frac58	⅝

Brown	LOB	SGML	XML/Unicode
		&frac78	⅞
		&sup1	¹
		&sup2	²
		&sup3	³
*A, *B, *C, ...	A, B, C, ...	A, B, C, ...	A, B, C, ...
A, B, C, ...	a, b, c, ...	a, b, c, ...	a, b, c, ...
		&AElig	Æ
	ae	&aelig	æ
		&eth	ð
		&OElig	Œ
	oe	&oelig	œ
	ss	&szlig	ß
		&THORN	Þ
	*?24	&thorn	þ
	*?26		ȝ
	l?2	&*l*acute	ÁáćÉéÍíĹĺńÓóŕŚśÚúýź
		&acute	´
		&abreve	ă
	l?10	&*l*caron	Ččď Ěěň Řř Šš ť Žž
	l?6	&*l*cedil	ÇçņŞşţ
	l?5	&*l*circ	ÂâĉÊêÎîÔôŝÛûŵŶŷ
		&circ	ˆ
		&zdot	ż
	l?3	&*l*grave	ÀàÈèìòù
		&grave	`
	l?1	&*l*macr	ĀāĒēīōū
		&*l*ogon	ąę
	l?15	&*l*ring	Ååů
		&dstrok	đ
	*?19	&hstrok	ħ
	L*?11	&Lstrok	Ł
	l*?11	&lstrok	ł
	O*?11	&Oslash	Ø
	o*?11	&oslash	ø
	l?4	&*l*tilde	ãÕõÑñ
		&tilde	˜
*l***D	*l*"	&*l*uml	ÄäËëÏïÖöÜüŸÿ
		&uml	¨
		&die	¨
*ZA	\15A		A
*YA	\15a	&agr	α
*ZB	\15B	&Bgr	Β
*YB	\15b	&bgr	β
*ZG	\15G	&Ggr	Γ
*YG	\15g	&ggr	γ
*ZD	\15D	&Dgr	Δ
*YD	\15d	&dgr	δ
*ZE	\15E	&Egr	Ε

Brown	LOB	SGML	XML/Unicode
*YE	\15e	&egr	ε
*ZZ	\15Z	&Zgr	Z
*YZ	\15z	&zgr	ζ
*ZH	\15E		H
*YH	\15e	&eegr	η
*ZJ	{15TH}	&THgr	Θ
*YJ	{15th}	&thgr	θ
*ZI	\15I		I
*YI	\15i	&igr	ι
*ZK	\15K		K
*YK	\15k	&kgr	κ
*ZL	\15L		Λ
*YL	\15l	&lgr	λ
*ZM	\15M	&Mgr	M
*YM	\15m	&mgr	μ
*ZN	\15N		N
*YN	\15n	&ngr	ν
*ZX	\15X		Ξ
*YX	\15x	&xgr	ξ
*ZO	\15O	&Ogr	O
*YO	\15o	&ogr	ο
*ZP	\15P	&Pgr	Π
*YP	\15p	&pgr	π
*ZR	\15R		P
*YR	\15r	&rgr	ρ
*ZS	\15S	&Sgr	Σ
*YS	\15s	&sgr	σ
*ZT	\15T		T
*YT	\15t	&tgr	τ
*ZU	\15U	&Ugr	Y
*YU	\15u	&ugr	υ
*ZF	\15F	&PHgr	Φ
*YF	\15f	&phgr	ϕ
*ZC	{15CH}		X
*YC	{15cH}	&khgr	χ
*ZY	{15PS}	&PSgr	Ψ
*YY	{15ps}	&psgr	ψ
*ZQ	\15O	&OHgr	Ω
*YQ	\15o	&ohgr	ω
		&ohm	Ω
**A	’	’	’
**C	:	:	:
**S	;	;	;
**I	?	?	?
**X	!	!	!
.	.	.	.
,	,	,	,
-	-	-	-

Brown	LOB	SGML	XML/Unicode
(	(	(	(
)	)	)	)
*	*/	&ast	*
=		&equals	=
**K		&percnt	%
+		&plus	+
+		&amp	& *for* &
		&sol	/
		&horbar	―
		&lowbar	_
		&dash	―
	-	&ndash	–
**-	*-	&mdash	—
	*?18	&ape	≈
		&bsol	\
		&bull	•
		&cent	¢
		&check	√
		&cir	°
		&commat	@
		&copy	©
	*?16	&darr	↓
*+0	*@	&deg	°
		&divide	÷
		&dollar	$
		&dtrif	▼
		&flat	♭
		&ft	′
		&ge	⩾
		&gt	> *or* >
		&Gt	≫
		&hearts	❤
		&hellip	…
		&iexcl	¡
	*?25	&infin	∞
		&ins	″
		&iquest	¿
		&larr	←
		&lcub	{
		&le	⩽
*(		&lsqb	[
		&lt	< *for* <
		&Lt	≪
		&micro	µ
	.	&middot	·
		&natur	♮
		&num	#
	*?14	&plusmn	±

Brown	LOB	SGML	XML/Unicode
	*+	&pound	£
	*?7	&prime	′
	*?8	&Prime	″
	*?9		‴
		&quot	" *or* "
	*?22	&radic	√
	*?23	&rarr	→
		&rcub	}
		&reg	®
*)		&rsqb	]
	*?13	&sect	§
		&sharp	♯
	*?17	&sim	~
		&shilling	/-
	x	&times	×
		&trade	™
		&verbar	|
		&yen	¥
**Q	*" *or* *’	&bquo	“
**U	**" *or* **’	&equo	”
**F	**[FORMULA**]	&formula	<gap desc="formula"/>
		&rehy	-
	^ sentence	<s>sentence</s>	
	~ included sent.		
	_ list	<list>list</list>	
	| paragraph	paragraph	
**N major heading **P	*< heading *>	<head>heading</head>	
**R minor heading **T			
	*o roman	<hi r=ro>roman</hi>	
*= italics *$	*1 italics	<hi r="it">italics</hi>	
**= bold face **$	*4 bold face	<hi r="bo">bold</hi>	
**(capitalised **)	*2 capitalised		
	**[comment **]	<!-- comment -->	

Appendix C.

Part-of-speech tags

The following table gives a brief overview of different sets of part-of-speech tags: *Brown* is the tagset used in the tagged Brown Corpus (Francis & Kučera 1964); *LOB*, the tagset used in the tagged London–Oslo/Bergen Corpus (Johansson *& al.* 1986); *C5*[30], the tagset used in the British National Corpus; *C2*[31], the tagset used by rasp; *C7*[32], an updated version of *C2*; and *W*, a simplified set of word classes (Burnard 2007) added to the last edition of the bnc. Only the correspondences between *W*, *C5* and *C7* are authoritative; the mapping between *C2* and *C7* should be fairly good given the similarity between the tagsets; the correspondence with *Brown* and *LOB* is only approximate, however, and there are many differences which do not appear in the synoptic table below.

W	C5	Brown	LOB	C2	C7	
SUBST	NN0	NN	NN		NN	common noun sg./pl.: *sheep, cod*
	NN0				NNJ	organisation: *department, council*
	NN0		NNU		NNU	unit of measurement: *in., cc.*
	NN1	NR			ND1	direction: *north, southeast*
	NN1	NN	NN		NN1	singular: *book, girl*
	NN1				NNT1	temporal noun sg.: *day, week, year*
	NN1		NNU		NNU1	unit, sg.: *inch, centimetre*
	NN2	NNS	NNS		NN2	pl. common noun: *books, girls*
	NN2				NNJ2	pl. org.: *governments, committees*
	NN2				NNT2	temporal pl.: days, weeks, years
	NN2		NNUS		NNU2	unit, pl.: *inches, centimetres*
	NN2	NRS			NPD2	weekday, pl.: *Sundays*
	NN2				NPM2	month, pl.: *Octobers*
	NP0	NN-TL	NPT	NNA	NNSA1	following title, sg.: *MA*
	NP0	NNS-TL	NPTS		NNSA2	following title, pl.
	NP0	NN-TL	NPT	NNB	NNSB	preceding title: *Rt. Hon.*
	NP0				NNSB1	preceding title, sg.: *Prof.*
	NP0	NNS-TL	NPTS		NNSB2	preceding title, pl.: *Messrs*
	NP0	NP	NPL		NNL1	locative, sg.: *Street, Bay*

[30]) http://ucrel.lancs.ac.uk/claws5tags.html

[31]) http://ucrel.lancs.ac.uk/claws2tags.html

[32]) http://ucrel.lancs.ac.uk/claws7tags.html

W	C5	Brown	LOB	C2	C7	
	NP0	NPS	NPLS	NNL2		locative, pl.: *islands, roads*
	NP0	NP		NP		proper noun: *Indies, Andes*
	NP0	NP		NP1		proper noun sg.: *London, Jane*
	NP0	NPS		NP2		proper noun pl.: *Browns, Reagans*
	NP0	NR		NPD1		weekday, sg.: Sunday
	NP0	NR		NPM1		month, sg.: October
	ZZ0	ZZ		ZZ1		Letter sg.: *A, a, B, π*
	ZZ0	ZZ		ZZ2		Letter pl.: *As, a's, Bs*
VERB	VBI	BE		VBI	VB0	*be* inf.
	VB0	BE		VB0	VB0	*be* subj./imp.
	VBB	BEM		VBM		*am*
	VBB	BER		VBR		*are, art*
	VBZ	BEZ		VBZ		*is*
	VBD	BED		VBDR		*were*
	VBD	BEDZ		VBDZ		*was*
	VHI	HV		VHI	VH0	*have* inf.
	VHB	HV		VH0	VH0	*have* fin.
	VHZ	HVZ		VHZ		*has*
	VHD	HVD		VHD		*had* past tense
	VDI	DO		VDI	VD0	*do* inf.
	VDB	DO		VD0	VD0	*do* fin.
	VDZ	DOZ		VDZ		*does*
	VDD	DOD		VDD		*did*
	VVI	VB		VVI	VV0	infinitive: *take, live*
	VVB	VB		VV0	VV0	finite base form: *take, live*
	VVZ	VBZ		VVZ		*-s* form: *takes, lives*
	VVD	VBD		VVD		past tense: *took, lived*
	VM0	MD		VM		modal: *can, could*
	VM0	MD		VMK		catenative: *ought, used (to)*
	BEG	BEG		VBG		*being*
	VHG	HVG		VHG		*having*
	VDG	VBG		VDG		*doing*
	VVG	VBG		VVG		*-ing* part.: *giving, working*
	VVG	VBG		VVGK		catenative: *(be) going (to)*
	VBN	BEN		VBN		*been*
	VHN	HVN		VHN		*had* past part.
	VDN	VBN		VDN		*done*
	VVN	VBN		VVN		past part.: *given, worked*
	VVN	VBN		VVNK		catenative: *(be) bound (to)*
ADJ	AJ0	JJS		JJ	JJ	semantic superlative: *chief, top, utmost*
	AJ0	JJ		JJ	JJ	general adjective
	AJ0	JJ		JJ	JA	predicative: *tantamount, asleep*
	AJ0	JJ	JJB	JJ	JB	attributive: *main, chief, utter*
	AJ0	JJ	JJ	JK		catenative: *(be) able, willing (to)*

W	C5	Brown	LOB	C2	C7	
	AJC	JJR		JJR	JJR	comparative: *older, better, bigger*
	AJC				JBR	attr. comp.: *upper, outer*
	AJC	JJT		JJT	JJT	superlative: *oldest, best, biggest*
	AJC				JBT	attr. sup.: utmost, uttermost
	CRD	CD	CD1	MC1		*one*
	CRD		CD	MC		cardinal number: *two, three*
	CRD			NNO		numeral noun: *dozen, thousand*
	CRD			MF		fraction: *quarters, two-thirds*
	CRD		CDS	NNO2		*hundreds, thousands*
	CRD			MC2		*tens, twenties*
	CRD		CD$	MCGE	MC$	genitive: *10's*
	CRD		CD-CD	MCMC	MC-MC	hyph'd number: *40-50, 1770-1827*
	ORD	OD		MD		ordinal numeral: *first, second*
	DT0	AP		DA		after-det.: *such, former, same*
	DT0			DA1		singular: *little, much*
	DT0			DA2		plural: *few, several, many*
	DT0			DAR	DAR	comparative: *more, less*
	DT0				DA2R	plural: *fewer*
	DT0			DAT		superlative: *most, least*
	DT0	ABN		DB		before-det.: *all, half*
	DT0	ABX		DB2		plural: *both*
	DT0	DTI		DD		determiner: *any, some*
	DT0	DT		DD1		singular: *this, that, another*
	DT0	DTS		DD2		plural: *these, those*
ADV	AV0		CS	BCL	BCS	before conj.: *in order (that), even (if)*
	AV0		TO		BTO	before inf. marker: *in order, so as (to)*
	AV0			RA		post nominal: *else, galore*
	AV0			REX		appos. introd.: *namely, viz, e.g.*
	AV0	QL		RG	RG	degree adv: *very, so, to*
	AV0	QLP			RGA	post-nominal/adv./adj.: *indeed, enough*
	AV0	QL		RGR		comparative degree adv.: *more, less*
	AV0			RGT		superlative degree adv.: *most, least*
	AV0	RN		RL		locative: *alongside, forward*
	AV0	RB		RR		general adverb
	AV0	RBR		RRR		comparative: *better, longer*
	AV0	RBT		RRT		superlative: *best, longest*
	AV0	RN		RT		nominal adv. of time: *now, tomorrow*
	AVP	RP		RP		prep. adv./particle: *in, up, about*
	AVP			RPK		catenative: *(be) about (to)*
	AVQ	WQL		RGQ		*wh-* degree adv.: *how*
	AVQ	WDT		RGQV		*wh-ever* degree adv.: *however*
	AVQ	WRB		RRQ		*wh-* general adv.: *where, when*
	AVQ	WDT		RRQV		*wh-ever* general adv.: *wherev., whenev.*
	XX0	*	XNOT	XX		*not, n't*

W	C5	Brown	LOB	C2	C7	
ART	AT0	AT	AT	AT		*the, no*
			AT1	AT1		*a, an*
PRON	DPS	PP$		APPGE	APP$	genitive pron.: *my, your, our*
	DTQ	WDT	WDT(R)	DDQ		*wh*-determiner: *which, what*
	DTQ			DDQV		*wh-ever* determiner: *whichever*
	DTQ	WP$	WP$(R)	DDQGE	DDQ$	*whose*
	EX0	EX		EX		*there* existential
	PNI	PN		PN		indef. pron.: *none*
	PNI			PN1		sing.: *anyone, everything*
	PNP	PPS	PP3	PPH1		*it*
	PNP		PP3A	PPHS1		*he, she*
	PNP	PPSS	PP1A	PPIS1		*I*
	PNP		PP1AS	PPIS2		*we*
	PNP		PP2	PPY		*you*
	PNP		PP3AS	PPHS2		*they*
	PNP	PPO	PP3O	PPHO1		*him, her*
	PNP		PP1O	PPIO1		*me*
	PNP		PP1OS	PPIO2		*us*
	PNP		PP3OS	PPHO2		*them*
	PNP	PP$$		PPGE	PP$	possessive pron.: *mine, ours*
	PNQ	WPS	WP(A)(R)	PNQS		*who*
	PNQ	WPO	WPO(R)	PNQO		*whom*
	PNQ	WPS	WPA(R)		PNQVS	*who(so)ever*
	PNQ	WPO	WPO(R)	PNQV	PNQVO	*whom(so)ever*
	PNQ	WP$	WP$(R)		PNQV$	*whosever*
	PNX	PPL		PNX1		*oneself*
	PNX			PPX1		reflexive sg.: *yourself, itself*
	PNX	PPLS		PPX2		reflexive pl.: *yourselves, ourselves*
PREP	PRF	IN		II		preposition
	PRP			IF		*for*
	PRP			IO		*of*
	PRP			IW		*with*
	TO0	TO		TO		*to* inf. marker

W	C5	Brown	LOB	C2	C7	
CONJ	CJC	CC		CC		coordinating: *and*, *or*
	CJC			CCB		*but*
	CJS			CS	CF	semi-coordinating: *so*, *then*, *yet*
	CJS				CS	subordinating: *if*, *because*, *unless*
	CJS	CS		CSA		*as*
	CJS			CSN		*than*
	CJS			CSW		*whether*
	CJT			CST		*that*
INTERJ	ITJ	UH		UH		interjection: *oh*, *yes*, *um*
UNC	UNC		&FO	FO	&FO	Formula
	UNC			FU		Unknown
	UNC	FW-	&FW	FW	&FW	Foreign word
STOP	POS	$		GE	$	Genitive marker
	PUN	.		.		Full stop
	PUN	;		;		Semicolon
	PUN	!		!		Exclamation mark
	PUN	?		?		Question mark
	PUN	,		,		Comma
	PUN	:		:		Colon
	PUL	(		(		Opening bracket
	PUR	)		)		Closing bracket
	PUN	—		—		Dash
	PUN		...	...		Ellipsis
	PUQ		*’ **’	"		Quotation mark

Appendix D.

XML listings

D.1. CLC mark-up from Chapter 3

```
<sent idx="247">
    <orig idx="247">Then <NS type="RD">some|a </NS> <NS type="SX">though|thought </NS>
                <NS type="IV">occured|occurred </NS> to me. </orig>
    <correct idx="247" partnum="1">
        <part num="1">
            <c_orig>Then <NS type="RD">some|a </NS> <NS type="SX">though|thought </NS>
                <NS type="IV">occured|occurred </NS> to me. </c_orig>
            <c_text>Then a thought occurred to me. </c_text>
            <c_rasp>
                <string><![CDATA[(|Then:1_RR| |a:2_AT1| |thought:3_NN1| |occur+ed:4_VVD|
                        |to:5_II| |I+:6_PPIO1| |.:7_.|) 1 ; (-9.204) ]]></string>
                <parse>
                    <tree>
                        <node rule="T/txt-sc1/-+">
                            <node rule="S/adv_s/-">
                                <node rule="AP/a1">
                                    <node rule="A1/a">
                                        <wnode>
                                            <pos>RR</pos>
                                            <word>Then</word>
                                        </wnode>
                                    </node>
                                </node>
                                <node rule="S/np_vp">
                                    <node rule="NP/det_n1">
                                        <wnode>
                                            <pos>AT1</pos>
                                            <word>a</word>
                                        </wnode>
                                        <node rule="N1/n">
                                            <wnode>
                                                <pos>NN1</pos>
```

```
                                        <word>thought</word>
                                    </wnode>
                                </node>
                            </node>
                            <node rule="V1/v_pp">
                                <wnode>
                                    <pos>VVD</pos>
                                    <word>occur+ed</word>
                                </wnode>
                                <node rule="PP/p1">
                                    <node rule="P1/p_np-pro">
                                        <wnode>
                                            <pos>II</pos>
                                            <word>to</word>
                                        </wnode>
                                        <wnode>
                                            <pos>PPIO1</pos>
                                            <word>I+</word>
                                        </wnode>
                                    </node>
                                </node>
                            </node>
                        </node>
                    </node>
                    <node rule="End-punct3/-">
                        <wnode>
                            <pos>.</pos>
                            <word>.</word>
                        </wnode>
                    </node>
                </node>
            </tree>
            <gr-list>
                <gr>
                    <gr-type>ncmod</gr-type>
                    <subtype>_</subtype>
                    <head><pos>VVD</pos><word>occur+ed</word></head>
                    <dep><pos>RR</pos><word>Then</word></dep>
                </gr>
                <gr>
                    <gr-type>ncsubj</gr-type>
                    <head><pos>VVD</pos><word>occur+ed</word></head>
                    <dep><pos>NN1</pos><word>thought</word></dep>
                    <init-gr>_</init-gr>
                </gr>
                <gr>
                    <gr-type>iobj</gr-type>
                    <head><pos>VVD</pos><word>occur+ed</word></head>
                    <dep><pos>II</pos><word>to</word></dep>
```

```
                                    </gr>
                                    <gr>
                                        <gr-type>dobj</gr-type>
                                        <head><pos>II</pos><word>to</word></head>
                                        <dep><pos>PPIO1</pos><word>I+</word></dep>
                                    </gr>
                                    <gr>
                                        <gr-type>det</gr-type>
                                        <head><pos>NN1</pos><word>thought</word></head>
                                        <dep><pos>AT1</pos><word>a</word></dep>
                                    </gr>
                                </gr-list>
                            </parse>
                        </c_rasp>
                    </part>
                </correct>
                <incorrect idx="247" partnum="1">
                    <part num="1">
                        <inc_orig>Then <NS type="RD">some|a </NS> <NS type="SX">though|thought </NS>
                            <NS type="IV">occured|occurred </NS> to me. </inc_orig>
                        <inc_text>Then some though occured to me. </inc_text>
                        <inc_rasp>
                            <string><![CDATA[(|Then:1_RR| |some:2_DD| |though:3_RR| |occur+ed:4_VVN|
                                |to:5_II| |I+:6_PPIO1| |.:7_.|) 1 ; (-9.738) ]]></string>
                            <parse>
                                <tree>
                                    <node rule="T/txt-sc1/-+">
                                        <node rule="S/adv_s/-">
                                            <node rule="AP/a1">
                                                <node rule="A1/a">
                                                    <wnode>
                                                        <pos>RR</pos>
                                                        <word>Then</word>
                                                    </wnode>
                                                </node>
                                            </node>
                                            <node rule="S/np_vp">
                                                <wnode>
                                                    <pos>DD</pos>
                                                    <word>some</word>
                                                </wnode>
                                                <node rule="V1/adv_vp">
                                                    <node rule="AP/a1">
                                                        <node rule="A1/a">
                                                            <wnode>
                                                                <pos>RR</pos>
                                                                <word>though</word>
                                                            </wnode>
                                                        </node>
```

```
                                    </node>
                                    <node rule="V1/v_pp">
                                        <wnode>
                                            <pos>VVN</pos>
                                            <word>occur+ed</word>
                                        </wnode>
                                        <node rule="PP/p1">
                                            <node rule="P1/p_np-pro">
                                                <wnode>
                                                    <pos>II</pos>
                                                    <word>to</word>
                                                </wnode>
                                                <wnode>
                                                    <pos>PPIO1</pos>
                                                    <word>I+</word>
                                                </wnode>
                                            </node>
                                        </node>
                                    </node>
                                </node>
                            </node>
                        </node>
                        <node rule="End-punct3/-">
                            <wnode>
                                <pos>.</pos>
                                <word>.</word>
                            </wnode>
                        </node>
                    </node>
                </tree>
                <gr-list>
                    <gr>
                        <gr-type>ncmod</gr-type>
                        <subtype>_</subtype>
                        <head><pos>VVN</pos><word>occur+ed</word></head>
                        <dep><pos>RR</pos><word>Then</word></dep>
                    </gr>
                    <gr>
                        <gr-type>ncsubj</gr-type>
                        <head><pos>VVN</pos><word>occur+ed</word></head>
                        <dep><pos>DD</pos><word>some</word></dep>
                        <init-gr>_</init-gr>
                    </gr>
                    <gr>
                        <gr-type>ncmod</gr-type>
                        <subtype>_</subtype>
                        <head><pos>VVN</pos><word>occur+ed</word></head>
                        <dep><pos>RR</pos><word>though</word></dep>
                    </gr>
```

```
                                        <gr>
                                            <gr-type>iobj</gr-type>
                                            <head><pos>VVN</pos><word>occur+ed</word></head>
                                            <dep><pos>II</pos><word>to</word></dep>
                                        </gr>
                                        <gr>
                                            <gr-type>dobj</gr-type>
                                            <head><pos>II</pos><word>to</word></head>
                                            <dep><pos>PPIO1</pos><word>I+</word></dep>
                                        </gr>
                                    </gr-list>
                                </parse>
                            </inc_rasp>
                        </part>
    </incorrect>
</sent>
```

D.2. BNC mark-up from Chapter 3

D.2.1. Original mark-up

The example sentence as it appears in the BNC. (This particular sentence is not actually part of the corpus, but has been assembled from parts which are.)

```
<s n="1">
   <trunc><w c5="UNC" hw="any" pos="UNC">Any </w></trunc>
   <w  c5="PNI" hw="anyone" pos="PRON">anyone </w>
   <w c5="PNQ" hw="who" pos="PRON">who </w>
   <w c5="AJ0-VVN" hw="dissolved" pos="ADJ">dissolved </w>
   <mw c5="AV0">
         <w c5="AV0" hw="more" pos="ADV">more </w>
         <w c5="CJS" hw="than" pos="CONJ">than </w>
   </mw>
   <w c5="UNC" hw="½" pos="UNC">½ </w>
   <gap desc="formula"/>
   <w c5="PRP" hw="in" pos="PREP"> in </w>
   <w c5="NN1" hw="rivers/lakes" pos="SUBST">rivers/lakes </w>
   <w c5="VBZ" hw="be" pos="VERB">is</w>
   <w c5="XX0" hw="not" pos="ADV">n't </w>
   <w c5="VVG" hw="gon" pos="VERB">gon</w>
   <w c5="TO0" hw="na" pos="PREP">na </w>
   <w c5="VVI" hw="forget" pos="VERB">forget </w>
   <w c5="DPS" hw="his/her" pos="PRON">his </w>
   <w c5="NN1" hw="pilgrimage" pos="SUBST">pilgrimage</w>
   <c c5="PUN">, </c>
   <w c5="VVB-NN1" hw="y'know" pos="VERB">y'know</w>
   <c c5="PUN">.</c>
</s>
```

D.2.2. After parsing

The example sentence after parsing with the current version of RASP4UIMA. Added elements and attributes are printed in italics. Note that the attributes *n*, *rpos*, *lem* and *affix* are space-separated lists when tokens have been split; in particular, the *affix* attribute for *rivers/lakes* has the value "*+s +s*" with two spaces.

```
<s n="1">
  <trunc><w c5="UNC" hw="any" pos="UNC">Any </w></trunc>
  <w n="1" c5="PNI" hw="anyone" pos="PRON" rpos="PN1" lem="anyone">anyone </w>
  <w n="2" c5="PNQ" hw="who" pos="PRON" rpos="PNQS" lem="who">who </w>
  <w n="3" c5="AJ0-VVN" hw="dissolved" pos="ADJ"
    rpos="VVD" lem="dissolve" affix="+ed">dissolved </w>
  <mw c5="AV0">
    <w n="4" c5="AV0" hw="more" pos="ADV" rpos="DAR" lem="more">more </w>
    <w n="5" c5="CJS" hw="than" pos="CONJ" rpos="CSN" lem="than">than </w>
  </mw>
  <w n="6" c5="UNC" hw="½" pos="UNC" rpos="MC" lem="½">½ </w>
  <gap n="7" desc="formula" rpos="&amp;amp;FO" lem="[gap]"/>
  <w n="8" c5="PRP" hw="in" pos="PREP" rpos="II" lem="in"> in </w>
  <w n="9 10 11" c5="NN1" hw="rivers/lakes" pos="SUBST"
    rpos="NNL2 CC NN2" lem="river / lake" affix="+s +s">rivers/lakes </w>
  <w n="12" c5="VBZ" hw="be" pos="VERB" rpos="VBZ" lem="be" affix="+s">is</w>
  <w n="13" c5="XX0" hw="not" pos="ADV" rpos="XX" lem="not" affix="+">n't </w>
  <w n="14" c5="VVG" hw="gon" pos="VERB" rpos="VVN" lem="gon">gon</w>
  <w n="15" c5="TO0" hw="na" pos="PREP" rpos="TO" lem="na">na </w>
  <w n="16" c5="VVI" hw="forget" pos="VERB" rpos="VV0" lem="forget">forget </w>
  <w n="17" c5="DPS" hw="his/her" pos="PRON" rpos="APP$" lem="his">his </w>
  <w n="18" c5="NN1" hw="pilgrimage" pos="SUBST" rpos="NN1" lem="pilgrimage">pilgrimage</w>
  <c n="19" c5="PUN" rpos="," lem=",">, </c>
  <w n="20 21" c5="VVB-NN1" hw="y'know" pos="VERB"
    rpos="PPY VV0" lem="y' know">y'know</w>
  <c n="22" c5="PUN" rpos="." lem=".">.</c>
  <grlist parse="1" score="-40.848">
    <gr type="ncsubj" head="14" dep="1"/>
    <gr type="cmod" subtype="_" head="1" dep="3"/>
    <gr type="ncsubj" head="3" dep="2"/>
    <gr type="ncmod" subtype="_" head="3" dep="8"/>
    <gr type="xcomp" subtype="_" head="3" dep="4"/>
    <gr type="ncmod" subtype="_" head="4" dep="5"/>
    <gr type="dobj" head="5" dep="7"/>
    <gr type="ncmod" subtype="_" head="7" dep="6"/>
    <gr type="dobj" head="8" dep="10"/>
    <gr type="conj" head="10" dep="9"/>
    <gr type="conj" head="10" dep="11"/>
    <gr type="aux" head="14" dep="12"/>
    <gr type="ncmod" subtype="_" head="14" dep="13"/>
    <gr type="ta" subtype="end" head="14" dep="21"/>
    <gr type="xcomp" subtype="to" head="14" dep="16"/>
    <gr type="passive" head="14"/>
    <gr type="dobj" head="16" dep="18"/>
```

```
        <gr type="det" head="18" dep="17"/>
        <gr type="ncsubj" head="21" dep="20"/>
    </grlist>
</s>
```

D.3. From Chapter 7

D.3.1. example.xml

```
<!DOCTYPE learner SYSTEM "LNRC.dtd">
<learner>
        <head url="1865831">
                <candidate>
                        <exam>
                                <exam_code>0085</exam_code>
                                <exam_desc>Key English Test</exam_desc>
                                <exam_level>KET</exam_level> </exam>
                        <personnel>
                                <language>Spanish</language>
                                ... </personnel>
                        <scores_and_grades>
                                        ... </scores_and_grades> </candidate>
                <text>
                        <answer1>
                                <question_number>9</question_number>
                                <exam_score>4</exam_score>
                                <original_answer>
                                        <p>We can not stomach an other christmastime
                                            celebration with all kind of delicacys for sweet
                                            tooths.</p>
                                        <p>In the other hand, it would be also a sham too
                                            disappoint an hopefull childly heart.</p>
                                                </original_answer> </answer1> </text> </head>
        <head url="2845832">
                <candidate>
                        <exam>
                                <exam_code>0100</exam_code>
                                <exam_desc>First Certificate of English</exam_desc>
                                <exam_level>FCE</exam_level> </exam>
                        <personnel>
                                <language>Italian</language>
                                ... </personnel>
                        <scores_and_grades>
                                        ... </scores_and_grades> </candidate>
                <text>
                        <answer1>
                                <question_number>7</question_number>
                                <exam_score>5</exam_score>
```

```
<original_answer>
    <p>The cr&egrave;me br&ucirc;l&eacute;e costed
        &pound;18 at Mangiare &amp; Bere.</p>
</original_answer> </answer1> </text> </head> </learner>
```

D.3.2. example.xml.o

```
<?xml version="1.0"?>
<rasp>
    <p>We can not stomach an other christmastime celebration with all kind of delicacys for sweet
        tooths.</p>
    <p>In the other hand, it would be also a sham too disappoint an hopefull childly heart.</p>
    <p>The crème brûlée costed £18 at Mangiare &amp; Bere.</p>
</rasp>
```

D.3.3. example.xml.otag

```
<?xml version="1.0"?>
<rasp>
    <p>We <e t="RP 0rp"><i>can not</i><c>cannot</c></e> stomach
        <e t="RP 0rp"><i>an other</i><c>another</c></e> christmastime celebration with all
        <e t="AGN 0agn"><i>kind</i><c>kinds</c></e> of delicacys for sweet tooths.</p>
    <p><e t="ID 0id"><i>In the other</i><c>On the other</c></e> hand, it would
        <e t="W 0w"><i>be also</i><c>also be</c></e> a sham too disappoint an
        <e t="DJ 0dj"><i>hopefull</i><c>hopeful</c></e> childly heart.</p>
    <p>The crème brûlée <e t="IV 0iv"><i>costed</i><c>cost</c></e> £18 at Mangiare &amp; Bere.</p>
</rasp>
```

D.3.4. example.xml.2

```
<?xml version="1.0"?>
<rasp>
    <p>We cannot stomach another christmastime celebration with all kinds of delicacys for sweet
        tooths.</p>
    <p>On the other hand, it would also be a sham too disappoint an hopeful childly heart.</p>
    <p>The crème brûlée cost £18 at Mangiare &amp; Bere.</p>
</rasp>
```

D.3.5. example.xml.2token

```
<rasp>
    <p>
        <start-s n="0"/> ^ <w s='0' e='1'>We</w> <w s='3' e='8'>cannot</w>
```

```
        <w s='10' e='16'>stomach</w> <w s='18' e='24'>another</w> <w s='26' e='38'>christmastime</w>
        <w s='40' e='50'>celebration</w> <w s='52' e='55'>with</w> <w s='57' e='59'>all</w>
        <w s='61' e='65'>kinds</w> <w s='67' e='68'>of</w> <w s='70' e='78'>delicacys</w>
        <w s='80' e='82'>for</w> <w s='84' e='88'>sweet</w> <w s='90' e='95'>tooths</w>
        <w s='96' e='96'>.</w> ^^ <end-s n="96"/>
    </p>
    <p>
        <start-s n="0"/> ^ <w s='0' e='1'>On</w> <w s='3' e='5'>the</w> <w s='7' e='11'>other</w>
        <w s='13' e='16'>hand</w> <w s='17' e='17'>,</w> <w s='19' e='20'>it</w>
        <w s='22' e='26'>would</w> <w s='28' e='31'>also</w> <w s='33' e='34'>be</w>
        <w s='36' e='36'>a</w> <w s='38' e='41'>sham</w> <w s='43' e='45'>too</w>
        <w s='47' e='56'>disappoint</w> <w s='58' e='59'>an</w> <w s='61' e='67'>hopeful</w>
        <w s='69' e='75'>childly</w> <w s='77' e='81'>heart</w> <w s='82' e='82'>.</w> ^^
        <end-s n="82"/>
    </p>
    <p>
        <start-s n="0"/> ^ <w s='0' e='2'>The</w> <w s='4' e='8'>crème</w> <w s='10' e='15'>brûlée</w>
        <w s='17' e='20'>cost</w> <w s='22' e='22'>£</w> <w s='23' e='24'>18</w>
        <w s='26' e='27'>at</w> <w s='29' e='36'>Mangiare</w> <w s='38' e='38'>&amp;amp;</w>
        <w s='40' e='43'>Bere</w> <w s='44' e='44'>.</w> ^^<end-s n="44"/>
    </p>
</rasp>
```

D.3.6. example.xml.2tag

```
<?xml version="1.0"?>
<rasp>
    <p>We <e t="RP 0rp"><i>can not</i><c>cannot</c></e> stomach
        <e t="RP 0rp"><i>an other</i><c>another</c></e>
        <e t="RP 2rp">i>christmastime</i><c>Christmastime</c></e> celebration with all
        <e t="AGN 0agn"><i>kind</i><c>kinds</c></e> of
        <e t="IN 2in"><i>delicacys</i><c>delicacies</c></e> for sweet
        <e t="IN 2in"><i>tooths</i><c>teeth</c></e>.</p>
    <p><e t="ID 0id"><i>In the other</i><c>On the other</c></e> hand, it would
        <e t="W 0w"><i>be also</i><c>also be</c></e> a sham too disappoint an
        <e t="DJ 0dj"><i>hopefull</i><c>hopeful</c></e> <e t="S 2s"><i>childly</i><c>(¿?)</c></e>
        heart.</p>
    <p>The crème brûlée <e t="IV 0iv"><i>costed</i><c>cost</c></e> £18 at Mangiare &amp;amp; Bere.</p>
</rasp>
```

D.3.7. example.xml.3

```
<?xml version="1.0"?>
<rasp>
    <p>We cannot stomach another Christmastime celebration with all kinds of delicacies for sweet
        teeth.</p>
```

```
    <p>On the other hand, it would also be a sham too disappoint an hopeful (¿?) heart.</p>
    <p>The crème brûlée cost £18 at Mangiare &amp;amp; Bere.</p>
</rasp>
```

D.3.8. example.xml.3pos

```
<?xml version="1.0"?>
<rasp>
    <p>
        <start-s n="0"/>
        <sentence num='1'>
            <lemma-list>
                <lemma s='0' e='1' lem='We' num='1' wnum='1' pos='PPIS2'/>
                <lemma s='3' e='8' lem='cannot' num='2' wnum='2' pos='VM'/>
                <lemma s='10' e='16' lem='stomach' num='3' wnum='3' pos='VV0'/>
                <lemma s='18' e='24' lem='another' num='4' wnum='4' pos='DD1'/>
                <lemma s='26' e='38' lem='Christmastime' num='5' wnum='5' pos='NN1'/>
                <lemma s='40' e='50' lem='celebration' num='6' wnum='6' pos='NN1'/>
                <lemma s='52' e='55' lem='with' num='7' wnum='7' pos='IW'/>
                <lemma s='57' e='59' lem='all' num='8' wnum='8' pos='DB2'/>
                <lemma s='61' e='65' lem='kinds' num='9' wnum='9' pos='NN2'/>
                <lemma s='67' e='68' lem='of' num='10' wnum='10' pos='IO'/>
                <lemma s='70' e='79' lem='delicacies' num='11' wnum='11' pos='NN2'/>
                <lemma s='81' e='83' lem='for' num='12' wnum='12' pos='IF'/>
                <lemma s='85' e='89' lem='sweet' num='13' wnum='13' pos='JJ'/>
                <lemma s='91' e='95' lem='teeth' num='14' wnum='14' pos='NN2'/>
                <lemma s='96' e='96' lem='.' num='15' wnum='15' pos='.'/>
            </lemma-list>
        </sentence>
        <end-s n="96"/>
    </p>
    <p>
        <start-s n="0"/>
        <sentence num='2'>
            <lemma-list>
                <lemma s='0' e='1' lem='On' num='1' wnum='1' pos='II'/>
                <lemma s='3' e='5' lem='the' num='2' wnum='2' pos='AT'/>
                <lemma s='7' e='11' lem='other' num='3' wnum='3' pos='JB'/>
                <lemma s='13' e='16' lem='hand' num='4' wnum='4' pos='NN1'/>
                <lemma s='17' e='17' lem=',' num='5' wnum='5' pos=','/>
                <lemma s='19' e='20' lem='it' num='6' wnum='6' pos='PPH1'/>
                <lemma s='22' e='26' lem='would' num='7' wnum='7' pos='VM'/>
                <lemma s='28' e='31' lem='also' num='8' wnum='8' pos='RR'/>
                <lemma s='33' e='34' lem='be' num='9' wnum='9' pos='VB0'/>
                <lemma s='36' e='36' lem='a' num='10' wnum='10' pos='AT1'/>
                <lemma s='38' e='41' lem='sham' num='11' wnum='11' pos='NN1'/>
                <lemma s='43' e='45' lem='too' num='12' wnum='12' pos='RG'/>
                <lemma s='47' e='56' lem='disappoint' num='13' wnum='13' pos='VV0'/>
```

```
                        <lemma s='58' e='59' lem='an' num='14' wnum='14' pos='AT1'/>
                        <lemma s='61' e='67' lem='hopeful' num='15' wnum='15' pos='JJ'/>
                        <lemma s='69' e='72' lem='(¿?)' num='16' wnum='16' pos='NN1'/>
                        <lemma s='74' e='78' lem='heart' num='17' wnum='17' pos='NN1'/>
                        <lemma s='79' e='79' lem='.' num='18' wnum='18' pos='.'/>
                </lemma-list>
        </sentence>
        <end-s n="79"/>
    </p>
    <p>
        <start-s n="0"/>
        <sentence num='3'>
                <lemma-list>
                        <lemma s='0' e='2' lem='The' num='1' wnum='1' pos='AT'/>
                        <lemma s='4' e='8' lem='crème' num='2' wnum='2' pos='NN1'/>
                        <lemma s='10' e='15' lem='brûlée' num='3' wnum='3' pos='NN1'/>
                        <lemma s='17' e='20' lem='cost' num='4' wnum='4' pos='VVD'/>
                        <lemma s='22' e='22' lem='£' num='5' wnum='5' pos='NNU'/>
                        <lemma s='23' e='24' lem='18' num='6' wnum='6' pos='MC'/>
                        <lemma s='26' e='27' lem='at' num='7' wnum='7' pos='II'/>
                        <lemma s='29' e='36' lem='Mangiare' num='8' wnum='8' pos='NP1'/>
                        <lemma s='38' e='38' lem='&amp;amp;' num='9' wnum='9' pos='CC'/>
                        <lemma s='40' e='43' lem='Bere' num='10' wnum='10' pos='NP1'/>
                        <lemma s='44' e='44' lem='.' num='11' wnum='11' pos='.'/>
                </lemma-list>
        </sentence>
        <end-s n="44"/>
    </p>
</rasp>
```

D.3.9. example.xml.3tag

```
<?xml version="1.0"?>
<rasp>
    <p>We <e t="RP 0rp"><i>can not</i><c>cannot</c></e> stomach
        <e t="RP 0rp"><i>an other</i><c>another</c></e>
        <e t="RP 2rp"><i>christmastime</i><c>Christmastime</c></e> celebration with all
        <e t="AGN 0agn"><i>kind</i><c>kinds</c></e> of
        <e t="IN 2in"><i>delicacys</i><c>delicacies</c></e> for sweet
        <e t="IN 2in"><i>tooths</i><c>teeth</c></e>.</p>
    <p><e t="ID 0id"><i>In the other</i><c>On the other</c></e> hand, it would
        <e t="W 0w"><i>be also</i><c>also be</c></e> a sham too disappoint
        <e t="FD 3fd"><i>an</i><c>a</c></e> <e t="DJ 0dj"><i>hopefull</i><c>hopeful</c></e>
        <e t="S 2s"><i>childly</i><c>(¿?)</c></e> heart.</p>
    <p>The crème brûlée <e t="IV 0iv"><i>costed</i><c>cost</c></e> £18 at Mangiare &amp;amp; Bere.</p>
</rasp>
```

D.3.10. example.xml.5

```
<?xml version="1.0"?>
<rasp>
    <p>We cannot stomach another Christmastime celebration with all kinds of delicacies for sweet
        teeth.</p>
    <p>On the other hand, it would also be a sham too disappoint a hopeful (¿?) heart.</p>
    <p>The crème brûlée cost £18 at Mangiare &amp;amp; Bere.</p>
</rasp>
```

D.3.11. example.xml.5rasp

```
<?xml version="1.0"?>
<rasp>
    <p>
        <start-s n="0"/>
        <sentence num='1'>
            <lemma-list>
                <lemma s='0' e='1' lem='We' num='1' wnum='1' pos='PPIS2'/>
                <lemma s='3' e='8' lem='cannot' num='2' wnum='2' pos='VM'/>
                <lemma s='10' e='16' lem='stomach' num='3' wnum='3' pos='VV0'/>
                <lemma s='18' e='24' lem='another' num='4' wnum='4' pos='DD1'/>
                <lemma s='26' e='38' lem='Christmastime' num='5' wnum='5' pos='NN1'/>
                <lemma s='40' e='50' lem='celebration' num='6' wnum='6' pos='NN1'/>
                <lemma s='52' e='55' lem='with' num='7' wnum='7' pos='IW'/>
                <lemma s='57' e='59' lem='all' num='8' wnum='8' pos='DB2'/>
                <lemma s='61' e='65' lem='kind' affix='s' num='9' wnum='9' pos='NN2'/>
                <lemma s='67' e='68' lem='of' num='10' wnum='10' pos='IO'/>
                <lemma s='70' e='79' lem='delicacy' affix='s' num='11' wnum='11' pos='NN2'/>
                <lemma s='81' e='83' lem='for' num='12' wnum='12' pos='IF'/>
                <lemma s='85' e='89' lem='sweet' num='13' wnum='13' pos='JJ'/>
                <lemma s='91' e='95' lem='tooth' affix='s' num='14' wnum='14' pos='NN2'/>
                <lemma s='96' e='96' lem='.' num='15' wnum='15' pos='.'/>
            </lemma-list>
            <nbest-parses num='1'>
                <parse-set pnum='1' score='-19.371'>
                    <gr-list>
                        <gr type='ncsubj' head='3' dep='1'></gr>
                        <gr type='aux' head='3' dep='2'></gr>
                        <gr type='iobj' head='3' dep='7'></gr>
                        <gr type='dobj' head='3' dep='6'></gr>
                        <gr type='dobj' head='7' dep='9'></gr>
                        <gr type='iobj' head='9' dep='10'></gr>
                        <gr type='dobj' head='10' dep='11'></gr>
                        <gr type='ncmod' subtype='_' head='11' dep='12'></gr>
                        <gr type='dobj' head='12' dep='14'></gr>
                        <gr type='ncmod' subtype='_' head='14' dep='13'></gr>
                        <gr type='ncmod' subtype='part' head='9' dep='8'></gr>
```

```
                                        <gr type='det' head='6' dep='4'></gr>
                                        <gr type='ncmod' subtype='_' head='6' dep='5'></gr>
                                    </gr-list>
                                </parse-set>
                        </nbest-parses>
                </sentence>
                <end-s n="96"/>
        </p>
        <p>
                <start-s n="0"/>
                <sentence num='2'>
                        <lemma-list>
                                <lemma s='0' e='1' lem='On' num='1' wnum='1' pos='II'/>
                                <lemma s='3' e='5' lem='the' num='2' wnum='2' pos='AT'/>
                                <lemma s='7' e='11' lem='other' num='3' wnum='3' pos='JB'/>
                                <lemma s='13' e='16' lem='hand' num='4' wnum='4' pos='NN1'/>
                                <lemma s='17' e='17' lem=',' num='5' wnum='5' pos=','/>
                                <lemma s='19' e='20' lem='it' num='6' wnum='6' pos='PPH1'/>
                                <lemma s='22' e='26' lem='would' num='7' wnum='7' pos='VM'/>
                                <lemma s='28' e='31' lem='also' num='8' wnum='8' pos='RR'/>
                                <lemma s='33' e='34' lem='be' num='9' wnum='9' pos='VB0'/>
                                <lemma s='36' e='36' lem='a' num='10' wnum='10' pos='AT1'/>
                                <lemma s='38' e='41' lem='sham' num='11' wnum='11' pos='NN1'/>
                                <lemma s='43' e='45' lem='too' num='12' wnum='12' pos='RG'/>
                                <lemma s='47' e='56' lem='disappoint' num='13' wnum='13' pos='VV0'/>
                                <lemma s='58' e='58' lem='a' num='14' wnum='14' pos='AT1'/>
                                <lemma s='60' e='66' lem='hopeful' num='15' wnum='15' pos='JJ'/>
                                <lemma s='68' e='71' lem='(¿?)' num='16' wnum='16' pos='NN1'/>
                                <lemma s='73' e='77' lem='heart' num='17' wnum='17' pos='NN1'/>
                                <lemma s='78' e='78' lem='.' num='18' wnum='18' pos='.'/>
                        </lemma-list>
                        <nbest-parses num='1'>
                                <parse-set pnum='1' score='-20.003'>
                                        <gr-list>
                                                <gr type='ncmod' subtype='_' head='9' dep='1'></gr>
                                                <gr type='ncsubj' head='9' dep='6'></gr>
                                                <gr type='ncmod' subtype='_' head='9' dep='8'></gr>
                                                <gr type='aux' head='9' dep='7'></gr>
                                                <gr type='ccomp' subtype='_' head='9' dep='13'></gr>
                                                <gr type='ncsubj' head='13' dep='11'></gr>
                                                <gr type='ncmod' subtype='_' head='13' dep='12'></gr>
                                                <gr type='dobj' head='13' dep='17'></gr>
                                                <gr type='det' head='17' dep='14'></gr>
                                                <gr type='ncmod' subtype='_' head='17' dep='15'></gr>
                                                <gr type='ncmod' subtype='_' head='17' dep='16'></gr>
                                                <gr type='det' head='11' dep='10'></gr>
                                                <gr type='dobj' head='1' dep='4'></gr>
                                                <gr type='det' head='4' dep='2'></gr>
                                                <gr type='ncmod' subtype='_' head='4' dep='3'></gr>
```

```
                                    </gr-list>
                                </parse-set>
                        </nbest-parses>
                </sentence>
                <end-s n="78"/>
        </p>
        <p>
                <start-s n="0"/>
                <sentence num='3'>
                        <lemma-list>
                                <lemma s='0' e='2' lem='The' num='1' wnum='1' pos='AT'/>
                                <lemma s='4' e='8' lem='crème' num='2' wnum='2' pos='NN1'/>
                                <lemma s='10' e='15' lem='brûlée' num='3' wnum='3' pos='NN1'/>
                                <lemma s='17' e='20' lem='cost' num='4' wnum='4' pos='VVD'/>
                                <lemma s='22' e='22' lem='£' num='5' wnum='5' pos='NNU'/>
                                <lemma s='23' e='24' lem='18' num='6' wnum='6' pos='MC'/>
                                <lemma s='26' e='27' lem='at' num='7' wnum='7' pos='II'/>
                                <lemma s='29' e='36' lem='Mangiare' num='8' wnum='8' pos='NP1'/>
                                <lemma s='38' e='38' lem='&amp;amp;amp;' num='9' wnum='9' pos='CC'/>
                                <lemma s='40' e='43' lem='Bere' num='10' wnum='10' pos='NP1'/>
                                <lemma s='44' e='44' lem='.' num='11' wnum='11' pos='.'/>
                        </lemma-list>
                        <nbest-parses num='1'>
                                <parse-set pnum='1' score='-19.326'>
                                        <gr-list>
                                                <gr type='ncsubj' head='4' dep='3'></gr>
                                                <gr type='iobj' head='4' dep='7'></gr>
                                                <gr type='dobj' head='4' dep='5'></gr>
                                                <gr type='dobj' head='7' dep='9'></gr>
                                                <gr type='conj' head='9' dep='8'></gr>
                                                <gr type='conj' head='9' dep='10'></gr>
                                                <gr type='ncmod' subtype='num' head='5' dep='6'></gr>
                                                <gr type='det' head='3' dep='1'></gr>
                                                <gr type='ncmod' subtype='_' head='3' dep='2'></gr>
                                        </gr-list>
                                </parse-set>
                        </nbest-parses>
                </sentence>
                <end-s n="44"/>
        </p>
</rasp>
```

D.3.12. example.xml.pre/KET-Spanish-1865831.xml

```
<head url="1865831">
        <candidate>
                <exam>
                        <exam_code>0085</exam_code>
```

```
                    <exam_desc>Key English Test</exam_desc>
                    <exam_level>KET</exam_level>
                </exam>
                <personnel>
                        <language>Spanish</language>
                        ...
                </personnel>
                <scores_and_grades>
                                ...
                </scores_and_grades>
        </candidate>
        <text>
                <answer1>
                        <question_number>9</question_number>
                        <exam_score>4</exam_score>
                        <original_answer>
                                        <p>We <e t="RP 0rp"><i>can not</i><c>cannot</c></e> stomach
                                            <e t="RP 0rp"><i>an other</i><c>another</c></e>
                                            <e t="RP 2rp"><i>christmastime</i><c>Christmastime</c></e>
                                            celebration with all
                                            <e t="AGN 0agn"><i>kind</i><c>kinds</c></e> of
                                            <e t="IN 2in"><i>delicacys</i><c>delicacies</c></e> for sweet
                                            <e t="IN 2in"><i>tooths</i><c>teeth</c></e>.</p>
                                        <p><e t="ID 0id"><i>In the other</i><c>On the other</c></e>
                                            hand, it would <e t="W 0w"><i>be also</i><c>also be</c></e>
                                            a sham too disappoint <e t="FD 3fd"><i>an</i><c>a</c></e>
                                            <e t="DJ 0dj"><i>hopefull</i><c>hopeful</c></e>
                                            <e t="S 2s"><i>childly</i><c>(¿?)</c></e> heart.</p>
                        </original_answer>
                </answer1>
        </text>
</head>
```

D.3.13. example.xml.pre/FCE-Italian-2845832.xml

```
<head url="2845832">
        <candidate>
                <exam>
                        <exam_code>0100</exam_code>
                        <exam_desc>First Certificate of English</exam_desc>
                        <exam_level>FCE</exam_level>
                </exam>
                <personnel>
                        <language>Italian</language>
                        ...
                </personnel>
                <scores_and_grades>
                                ...
```

```
                </scores_and_grades>
        </candidate>
        <text>
                <answer1>
                        <question_number>7</question_number>
                        <exam_score>5</exam_score>
                        <original_answer>
                                        <p>The crème brûlée <e t="IV 0iv"><i>costed</i><c>cost</c></e>
                                            £18 at Mangiare &amp;amp; Bere.</p>
                        </original_answer>
                </answer1>
        </text>
</head>
```

References

Aarts Bas (2007). *Syntactic Gradience: The Nature of Grammatical Indeterminacy*. Oxford: OUP.

Académie Françoise (1694). *Le Dictionnaire de l'Académie Françoise: Dédié au Roy*. Paris: Coignard.

Accademia della Crusca (1612). *Vocabolario degli Accadmici della Crusca*. Consulted edition: Venice: Hertz, 1686.

Ahmad Farooq & Kondrak Grzegorz (2005). 'Learning a spelling error model from search query logs'. In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, Canada, pp. 955–962.

Aikawa Takako (2008). 'That was then, this is now'. Invited talk at the Workshop on Natural Language Processing resources, algorithms and tools for authoring aids, Sixth International Conference on Language Resources and Evaluation (LREC), Marrakech.

Allén Sture (1992). 'Opening address'. In: Jan Svartvik (ed.), *Directions in Corpus Linguistics: Proceedings of Nobel Symposium 82, Stockholm, 4–8 August 1991*. Trends in Linguistics: Studies and Monographs, vol. 65. Berlin: Mouton de Gruyter, pp. 1–3.

Andersen Øistein E. (2006). 'Grammatical error detection'. Master's thesis, Computer Lab., University of Cambridge.

Andrews Avery D. (1990). *Modern Icelandic Syntax*. Syntax and Semantics, vol. 24. San Diego, United States: Academic Press.

Aristotle (*Int.*). Περὶ ἑρμηνείας / *De Interpretatione*. Consulted edition: L. Minio-Paluello (ed.), *Aristotelis Categoriæ et liber De interpretatione*, Scriptorum classicorum bibliotheca Oxoniensis. Oxford: Clarendon, 1949.

Aristotle (*Rhet.*). Τέχνη ῥητορική / *Ars Rhetorica*. Consulted edition: Médéric Dufour (ed.), *Aristote: Rhétorique*. Paris: Les Belles Lettres, 1932.

Atwell Eric Steven (1987). 'How to detect grammatical errors in a text without parsing it'. In: *Proceedings of the third conference of the European chapter of the Association for Computational Linguistics*, Copenhagen, pp. 38–45.

Bate W. Jackson (1978). *Samuel Johnson*. London: Chatto & Windus.

Bender Emily M., Flickinger Dan, Oepen Stephan, Walsh Annemarie & Baldwin Timothy (2004). 'Arboretum: Using a precision grammar for grammar checking in CALL'. In: *Proceedings of the InSTIL/ICALL Symposium on Computer Assisted Learning: NLP and Speech Technologies in Advanced Language Learning Systems*, Venice, pp. 83–86.

Bigert Johnny (2004). 'Probabilistic detection of context-sensitive spelling errors'. In: *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, pp. 1633–1636.

Bigert Johnny, Sjöbergh Jonas, Knutsson Ola & Sahlgren Magnus (2005). 'Unsupervised evaluation of parser robustness'. In: Alexander F. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing:*

6th International Conference, CICLing 2005, Mexico City, Mexico, February 13–19, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3406. Berlin: Springer, pp. 142–154.

Bley-Vroman Robert (1983). 'The comparative fallacy in interlanguage studies: The case of systematicity'. *Language Learning*, vol. 33, pp. 1–17.

Bloomfield Leonard (1933). *Language*. London: Allen & Unwin.

Bray Tim & Sperberg-McQueen C. M. (1996). 'Extensible markup language (XML): W3C working draft 14–Nov–96'. <http://www.w3.org/TR/WD-xml-961114>.

Brill Eric & Moore Robert C. (2000). 'An improved error model for noisy channel spelling correction'. In: *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, pp. 286–293.

Briscoe Ted (2006). 'An introduction to tag sequence grammars and the RASP system parser'. Technical report, nº 662, Computer Laboratory, University of Cambridge.

Briscoe Ted & Carroll John (2006). 'Evaluating the accuracy of an unlexicalized statistical parser on the PARC DepBank'. In: *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, Sydney, pp. 41–48.

Briscoe Ted, Carroll John & Watson Rebecca (2006). 'The second release of the RASP system'. In: *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, pp. 77–80.

Brockett Chris, Dolan William B. & Gamon Michael (2006). 'Correcting ESL errors using phrasal SMT techniques'. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Sydney, pp. 249–256.

Bullokar William (1586). *Pamphlet for Grammar*. London: Bollifant. Facs. edn: J. R. Turner (ed.), *The works of William Bullokar: Vol. 2, Pamphlet for grammar, 1586*, Leeds Texts and Monographs, new series, vol. 1, School of English, University of Leeds, 1980.

Burnard Lou (1995). *Users' Reference Guide for the British National Corpus, version 1.0*. Oxford: Oxford University Computing Services.

Burnard Lou (1999). 'Using SGML for linguistic analysis: The case of the BNC'. *Markup Languages: Theory & Practice*, vol. 1, nº 2, pp. 31–51.

Burnard Lou (2007). 'Reference guide for the British National Corpus (XML edition)'. <http://www.natcorp.ox.ac.uk/XMLedition/URG/.

Burt Marina K. & Kiparsky Carol (1972). *The Gooficon: A repair manual for English*. Rowley, United States: Newbury House.

Cawdrey Robert (1604). *A Table Alphabeticall*. London: Weaver. Facs. edn: Amsterdam: Theatrum Orbis Terrarum, 1970.

Chen Chia-Yin, Hsien-Chin Liou & Chang Jason S. (2006). 'FAST: An automatic generation system for grammar tests'. In: *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney, pp. 1–4.

Chodorow Martin & Leacock Claudia (2000). 'An unsupervised method for detecting grammatical errors'. In: *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle, pp. 140–147.

Chomsky Noam (1957). *Syntactic Structures*. Janua Linguarum, series minor, vol. 4. The Hague: Mouton.

Chomsky Noam (1961). 'Some methodological remarks on generative grammar'. *Word: Journal of the Linguistic Circle of New York*, vol. 17, pp. 219–239.

Chomsky Noam (1962). 'A transformational approach to syntax'. In: A. A. Hill (ed.), *Third Texas Conference on Problems of Linguistic Analysis in English, May 9–12, 1958*. Studies in American English. Austin, Texas: University of Texas, pp. 124–169.

Clément Lionel & Villemonte de la Clergerie Éric (2005). 'MAF: A morphosyntactic annotation framework'. In: *Proceedings of the Second Language & Technology Conference*, Poznań, pp. 90–94.

Copestake Ann (2005). 'Investigating adjective denotation and collocation'. Invited talk at the Third International Workshop on Generative Approaches to the Lexicon, Geneva.

Corder S. P. (1971). 'Idiosyncratic dialects and error analysis'. *International Review of Applied Linguistics in Language Teaching (IRAL)*, vol. 9, nº. 2, pp. 147–160.

Corder S. Pit (1974). 'Error analysis'. In: J. P. B. Allen & S. Pit Corder (eds), *Techniques in Applied Linguisics*. The Edinburgh Course in Applied Linguistics. London: Oxford University Press.

Cortes Corinna & Vapnik Vladimir (1995). 'Support-vector networks'. *Machine Learning*, vol. 20, nº. 3, pp. 273–297.

Dagneaux Estelle, Denness Sharon & Granger Sylviane (1998). 'Computer-aided error analysis'. *System: The International Journal of Educational Technology and Language Learning Systems*, vol. 26, nº. 2, pp. 163–174.

Damerau Fred J. (1964). 'A technique for computer detection and correction of spelling errors'. *Communications of the ACM*, vol. 7, nº. 3, pp. 171–176.

De Felice Rachele & Pulman Stephen G. (2007). 'Automatically acquiring models of preposition use'. In: *Proceedings of the 4th ACL-SIGSEM Workshop on Prepositions*, Prague, pp. 45–50.

De Felice Rachele & Pulman Stephen G. (2008). 'A classifier-based approach to preposition and determiner error correction in L2 English'. In: *Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, pp. 169–176.

De Felice Rachele & Pulman Stephen G. (2009). 'Automatic detection of preposition errors in learner writing'. *CALICO Journal*, vol. 26, nº. 3, pp. 512–528.

Deorowicz Sebastian & Ciura Marcin G. (2005). 'Correcting spelling errors by modelling their causes'. *International Journal of Applied Mathematics and Computer Science*, vol. 15, nº. 2, pp. 275–285.

DeRose Steven (2004). 'Markup overlap: A review and a horse'. In: *Proceedings of Extreme Markup Languages*, Montréal.

Di Iorio Angelo, Peroni Silvio & Vitali Fabio (2009). 'Towards markup support for full GODDAGs and beyond: the EARMARK approach'. In: *Proceedings of Balisage: The Markup Conference 2009*, Montréal.

Díaz-Negrillo Ana & Fernández-Domínguez Jesús (2006). 'Error tagging systems for learner corpora'. *Revista Española de Lingüística Aplicada*, vol. 19, pp. 83–102.

Dickinson M. (2006). 'Writers' aids'. In: Keith Brown (ed.), *Encyclopedia of Language & Linguistics*. 2nd edn, Oxford: Elsevier, pp. 673–679.

Dinneen Francis P. (1967). *An Introduction to General Linguistics*. New York: Holt, Rinehart & Winston.

Dionysius of Thrace (*Ars Gr.*). Τέχνη γραμματική / *Ars Grammatica*. Consulted edition: Jean Lallot (ed.), *La grammaire de Denys le Thrace*, Sciences du langage. Paris: CNRS, 1989.

Dulay Heidi, Burt Marina & Krashen Stephen (1982). *Language 2*. New York: Oxford University Press.

Eglowstein Howard (1991). 'Can a grammar and style checker improve your writing?'. *BYTE*, vol. 16, nº. 8, pp. 238–242.

Ellis Rod & Barkhuizen Gary (2005). *Analysing Learner Language*. Oxford: OUP.

Ferrucci David & Lally Adam (2004). 'UIMA: An architectural approach to unstructured information processing in the corporate research environment'. *Natural Language Engineering*, vol. 10, nº. 3–4, pp. 327–348.

Fitikides T. J. (1936). *Common Mistakes in English: with Exercises*. Consulted: 6th edn, Harlow: Longman, 2002.

Fitzpatrick Eileen & Seegmiller M. S. (2004). In: Ulla Connor & Thomas A. Upton (eds), *The Montclair Electronic Language Database Project*. Language and Computers, vol. 52. Amsterdam: Rodopi, pp. 223–237.

Foster Jennifer (2004). 'Good reasons for noting bad grammar: Empirical investigations into the parsing of ungrammatical written English'. Ph.D. thesis, Trinity College, Dublin.

Foster Jennifer (2007). 'Treebanks gone bad: Parser evaluation and retraining using a treebank of ungrammatical sentences'. *International Journal on Document Analysis and Recognition*, vol. 10, nº 3, pp. 129–145.

Foster Jennifer & Andersen Øistein E. (2009). 'GenERRate: Generating errors for use in grammatical error detection'. In: *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*, Boulder, Colorado, pp. 82–90.

Foster Jennifer & Vogel Carl (2004). 'Parsing ill-formed text using an error grammar'. *Artificial Intelligence Review*, vol. 21, nº 3–4, pp. 269–291.

Fowler H. W. & F. G. (1906). *The King's English*. Oxford: Clarendon.

Francis W. N. & Kučera H. (1964). 'Brown corpus manual: Manual of information to accompany a standard corpus of present-day edited American English, for use with digital computers'. Consulted: 1979 edn, <http://khnt.aksis.uib.no/icame/manuals/brown/>.

Francis W. Nelson (1967). 'The Brown University Standard Corpus of English: Some implications for TESOL'. In: Betty Wallace Robinett (ed.), *On Teaching English to Speakers of Other Languages: Papers Read at the TESOL Conference, New York City, March 17–19 1966*. Washington: Institute of Language and Linguistics, Georgetown University, pp. 131–137.

Francis W. Nelson (1979). 'Problems of assembling and computerizing large corpora'. In: Henning Bergenholtz & Burkhard Schäder (eds), *Empirische Textwissenschaft*. Königstein: Scriptor, pp. 110–123. Consulted reprint in: Stig Johansson (ed.), *Computer Corpora in English Language Research*. Bergen: Norwegian Computing Centre for the Humanities, 1982, pp. 7–24.

Fürstenau Hagen & Lapata Mirella (2009). 'Graph alignment for semi-supervised semantic role labeling'. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, pp. 11–20.

Galletta Dennis F., Durcikova Alexandra, Everard Andrea & Jones Brian M. (2005). 'Does spell-checking software need a warning label?'. *Communications of the ACM*, vol. 48, nº 7, pp. 82–86.

Gamon Michael, Gao Jianfeng, Brockett Chris, Klementiev Alexandre, Dolan William B., Belenko Dmitriy & Vanderwende Lucy (2008). 'Using contextual speller techniques and language modeling for ESL error correction'. In: *Proceedings of the Third International Joint Conference on Natural Language Processing*, Hyderabad, India, pp. 449–456.

Garside Roger (1987). 'The CLAWS word-tagging system'. In: Roger Garside, Geoffrey Leech & Geoffrey Sampson (eds), *The Computational Analysis of English: A Corpus-based Approach*. London: Longman, pp. 30–41.

Goldfarb Charles F. (1990). *The SGML handbook*. Oxford: Clarendon.

Golding Andrew R. (1995). 'A Bayesian hybrid method for context-sensitive spelling correction'. In: *Proceedings of the Third Workshop on Very Large Corpora*, Boston, pp. 39–53.

Gram Lu & Buttery Paula (2009). 'A tutorial introduction to iLexIR Search'. Unpublished manuscript.

Granger Sylviane (2003). 'Error-tagged learner corpora and CALL: A promising synergy'. *CALICO Journal*, vol. 20, nº 3, pp. 465–480.

Granger Sylviane, Kraif Olivier, Ponton Claude, Antoniadis Georges & Zampa Virginie (2007). 'Integrating learner corpora and natural language processing: A crucial step towards reconciling technological sophistication and pedagogical effectiveness'. *ReCALL Journal*, vol. 19, nº 3, pp. 252–268.

Greene Barbara B. & Rubin Gerald M. (1971). 'Automatic grammatical tagging of English'. Technical report, Department of Linguistics, Brown University, Providence, United States.

Guthrie William (1971). *The Sophists*. Cambridge: CUP.

Hashemi Sylvana Sofkova, Cooper Robin & Andersson Robert (2003). 'Positive grammar checking: A finite state approach'. In: Alexander F. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing: 4th International Conference, CICLing 2003, Mexico City, Mexico, February 16–22, 2003, Proceedings*. Lecture Notes in Computer Science, vol. 2588. Berlin: Springer, pp. 635–646.

Hirst Graeme & Budanitsky Alexander (2005). 'Correcting real-word spelling errors by restoring lexical cohesion'. *Natural Language Engineering*, vol. 11, nº 1, pp. 87–111.

Hornby Albert Sydney (2005). *Oxford Advanced Learner's Dictionary*. 7th edn, Oxford: OUP.

Huang Jin Hu & Powers David (2001). 'Large scale experiments on correction of confused words'. In: *Proceedings of the 24th Australasian Conference on Computer Science*, Gold Coast, Australia, pp. 77–82.

Huddleston Rodney & Pullum Geoffrey K. (2002). *The Cambridge Grammar of the English Language*. Cambridge: CUP.

Ide Nancy, Baker Collin, Fellbaum Christiane, Fillmore Charles & Passonneau Rebecca (2008). 'MASC: the Manually Annotated Sub-Corpus of American English'. In: Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis & Daniel Tapias (eds), *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, Marrakech, pp. 2455–2460.

Ide Nancy & Suderman Keith (2007). 'GrAF: A graph-based format for linguistic annotations'. In: *Proceedings of the Linguistic Annotation Workshop*, Prague, pp. 1–8.

Ide Nancy & Suderman Keith (2009). 'Bridging the gaps: Interoperability for GrAF, GATE, and UIMA'. In: *Proceedings of the Third Linguistic Annotation Workshop, ACL-IJNLP 2009*, Singapore, pp. 27–34.

Isodore of Seville (*Orig.*). *Etymologiæ* or *Origines*. Consulted edition: Rudolph Beer (præfatus), *Isodori Etymologiæ: Codex Toletanus (nunc Matritensis) 15, 8, phototypice editus*, Codices Græci et Latini Photographice Depicti, vol. 13. Leyden: Sijthoff, 1909.

Izumi Emi, Uchimoto Kiyotaka & Isahara Hitoshi (2004). 'SST speech corpus of Japanese learners' English and automatic detection of learners' errors'. *ICAME Journal*, vol. 28, pp. 31–48.

Izumi Emi, Uchimoto Kiyotaka, Saiga Toyomi, Supnithi Thepchai & Isahara Hitoshi (2003). 'Automatic error detection in the Japanese learners' English spoken data'. In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, pp. 145–148.

James Carl (1998). *Errors in Language Learning and Use: Exploring Error Analysis*. London: Longman.

Jaynes E. T. (1957). 'Information theory and statistical mechanics'. *The Physical Review*, vol. 106, nº 4, pp. 620–630.

Joachims T. (2006). 'Training linear SVMs in linear time'. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, Philadelphia, United States, pp. 217–226.

Johansson Stig, Atwell Eric, Garside Roger & Leech Geoffrey (1986). *The Tagged LOB Corpus: Users' Manual*. Bergen: Norwegian Computing Centre for the Humanities.

Johansson Stig, Leech Goeffrey N. & Goodluck Helen (1978). *Manual of Information to accompany the Lancaster–Oslo/Bergen corpus of British English, for use with digital computers*. Oslo: Department of English, University of Oslo.

Johnson Samuel (1755). *A dictionary of the English language*. London: Knapton *& al.*

Kaeding F. W. (1897). *Häufigkeitswörterbuch der deutschen Sprache: Festgestellt durch einen Arbeitsanschuß der deutschen Stenographie-Systeme*. Steglitz, Germany: self-published.

Kemp C., Tenenbaum J. B., Griffiths T. L., Yamada T. & Ueda N. (2006). 'Learning systems of concepts with an infinite relational model'. In: *Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, pp. 381–388.

Kies Daniel (2008). 'Evaluating grammar checkers: A comparative ten-year study'. Presented at the 6th International Conference on Education and Information Systems, Technologies and Applications (EISTA), Orlando, United States. <http://papyr.com/hypertextbooks/grammar/gramchek.htm>.

Kistermann F. W. (1991). 'The invention and development of the Hollerith punched card: In commemoration of the 130th anniversary of the birth of Herman Hollerith and for the 100th anniversary of large scale data processing'. *Annals of the History of Computing*, vol. 13, nº 3, pp. 245–259.

Klein Sheldon & Simmons Robert F. (1963). 'A computational approach to grammatical coding of English words'. *Journal of the ACM*, vol. 10, nº 3, pp. 334–347.

Kohut Gary F. & Gorman Kevin J. (1995). 'The effectiveness of leading grammar/style software packages in analyzing business students' writing'. vol. 9, nº 3, pp. 341–361.

Kučera Henry (1967). *Computational analysis of present-day American English*. Providence, United States: Brown University Press.

Kučera Henry (1992). 'The odd couple: The linguist and the software engineer, the struggle for high quality computerized language aids'. In: Jan Svartvik (ed.), *Directions in Corpus Linguistics: Proceedings of Nobel Symposium 82, Stockholm, 4–8 August 1991*. Trends in Linguistics: Studies and Monographs, vol. 65. Berlin: Mouton de Gruyter.

Kukich Karen (1992). 'Techniques for automatically correcting words in text'. *ACM Computing Surveys*, vol. 24, nº 4, pp. 377–439.

Lee John & Seneff Stephanie (2008*a*). 'An analysis of grammatical errors in non-native speech in English'. In: *Proceedings of the IEEE Workshop on Spoken Language Technology*, Goa, India, pp. 89–92.

Lee John & Seneff Stephanie (2008*b*). 'Correcting misuse of verb forms'. In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Columbus, United States, pp. 174–182.

Leech Geoffrey (2005). 'Adding linguistic annotation'. In: Martin Wynne (ed.), *Developing Linguistic Corpora: a Guide to Good Practice*. Oxford: Oxbow Books, pp. 17–29. <http://ahds.ac.uk/linguistic-corpora/>.

Leech Geoffrey, Garside Roger & Bryant Michael (1994). 'CLAWS4: The tagging of the British National Corpus'. In: *Proceedings of the 15th International Conference on Computational Linguistics*, Kyoto, pp. 622–628.

Leech Geoffrey & Smith Nicholas (2000). 'Manual to accompany the British National Corpus (version 2) with improved word-class tagging'. <http://www.natcorp.ox.ac.uk/docs/bnc2postag_manual.htm>.

Lennon Paul (1991). 'Error: Some problems of definition, identification, and distinction'. *Applied Linguistics*, vol. 12, nº 2, pp. 180–196.

Lewin Ian (2007). 'BaseNPs that contain gene names: Domain specificity and genericity'. In: *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, Prague, pp. 163–170.

Lily William (anon.) (*c.* 1500). *Brevissima Institutio seu Ratio Grammatices Cognoscendæ*. Consulted edition: Geneva: Badius, 1557.

Liou Hsien-Chin (1991). 'Development of an English grammar checker: A progress report'. *CALICO Journal*, vol. 9, nº 1, pp. 57–70.

Littlestone Nick (1988). 'Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm'. *Machine Learning*, vol. 2, nº 4, pp. 285–318.

Lowth Robert (anon.) (1762). *A Short Introduction to English Grammar: With critical notes*. London: Millar & Dodsley.

Lüdeling Anke, Doolittle Seanna, Hirschmann Hagen, Schmidt Karin & Walter Maik (2008). 'Das Lernerkorpus Falko'. *Deutsch als Fremdsprache*, vol. 2, pp. 67–73.

Luhn H. P. (1960). 'Keyword-in-context index for technical literature'. *American Documentation*, vol. 11, nº 4, pp. 288–295.

Macdonald Nina H., Frase Lawrence T., Gingrich Patricia S. & Keenan Stacey A. (1982). 'The Writer's Workbench: Computer aids for text analysis'. *IEEE Transactions on Communications*, vol. 30, nº 1, pp. 105–110.

MacDonald Lightbound Penny (2005). 'An analysis of interlanguage errors in synchronous/asynchronous intercultural communication exchanges'. Ph.D. thesis, Universitat de València.

MacNeilage Peter F. (1964). 'Typing errors as clues to serial ordering mechanisms in language behaviour'. *Language and Speech*, vol. 7, nº 3, pp. 144–159.

Mangu Lidia & Brill Eric (1997). 'Automatic rule acquisition for spelling correction'. In: *Proceedings of the 14th International Conference on Machine Learning*, Nashville, United States, pp. 734–741.

Manning Christopher D. (2003). 'Probabilistic syntax'. In: Rens Bod, Jennifer Hay & Stefanie Jannedy (eds), *Probabilistic Linguistics*. Cambridge, United States: MIT Press, pp. 289–342.

Marcus Mitchell, Kim Grace, Marcinkiewicz Mary Ann, Macintyre Robert, Bies Ann, Ferguson Mark, Katz Karen & Schasberger Britta (1994). 'The Penn Treebank: Annotating predicate argument structure'. In: *Human Language Technology: Proceedings of a Workshop*, Plainsboro, United States, pp. 114–119.

Marcus Mitchell P., Santorini Beatrice & Marcinkiewicz Mary Ann (1993). 'Building a large annotated corpus of english: The Penn Treebank'. *Computational Linguistics*, vol. 19, nº 2, pp. 313–330.

Meilă Marina (2007). 'Comparing clusterings: An information based distance'. *Journal of Multivariate Analysis*, vol. 98, nº 5, pp. 873–895.

Michael Ian (1970). *English Grammatical Categories and the Tradition to 1800*. Cambridge: CUP.

Mishra Hari Mohan (1986). *A critique of Pāṇini's grammar*. Patna, India: Anupam Publications.

Mitkov Ruslan (2003). *The Oxford Handbook of Computational Linguistics*. Oxford: OUP.

Mitton R. (1986). 'A partial dictionary of English in computer-usable form'. *Literary and Linguistic Computing*, vol. 1, nº 4, pp. 214–215.

Mitton Roger, Hardcastle David & Pedler Jenny (2007). 'BNC! Handle with care! Spelling and tagging errors in the BNC'. In: *Proceedings of the Corpus Linguistics Conference*, Birmingham.

Navarro Daniel J., Griffiths Thomas L., Steyvers Mark & Lee Michael D. (2006). 'Modeling individual differences using Dirichlet processes'. *Journal of Mathematical Psychology*, vol. 50, nº 2, pp. 101–122.

Neal Radford M. (2000). 'Markov chain sampling methods for Dirichlet process mixture models'. *Journal of Computational and Graphical Statistics*, vol. 9, nº 2, pp. 249–265.

Nesselhauf Nadja (2004). 'Learner corpora and their potential for language teaching'. In: John McH. Sinclair (ed.), *How to use corpora in language teaching*. Amsterdam: Benjamins, pp. 125–152.

Newman Simon M., Swanson Rowena W. & Knowlton Kenneth (1959). 'A notation system for transliterating technical and scientific texts for use in data processing systems'. Technical report, nº 15, Office of Research and Development, Patent Office, Washington.

Nicholls Diane (2003). 'The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT'. In: Dawn Archer, Paul Rayson, Andrew Wilson & Tony McEnery (eds), *Proceedings of the Corpus Linguistics conference*. Technical Papers, vol. 16. Lancaster: University Centre For Computer Corpus Research on Lanugage, Lancaster University, pp. 572–581.

Nicholls Diane (2007). 'The <#S>compleat|complete</#S> learner corpus'. Unpublished manuscript.

Okanohara Daisuke & Tsujii Jun'ichi (2007). 'A discriminative language model with pseudo-negative samples'. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, pp. 73–80.

Page Brian (1990). 'Why do i have to get it right anyway?'. In: Brian Page (ed.), *What do you mean... it's wrong?* London: Centre for Information on Language Teaching and Research, pp. 102–106.

Peterson James L. (1980). 'Computer programs for detecting and correcting spelling errors'. *Communications of the ACM*, vol. 23, nº 12, pp. 676–687.

Pienemann Manfred (1992). 'COALA: A computational system for interlanguage analysis'. *Second Language Research*, vol. 8, nº 1, pp. 59–92.

Plato (*Crat.*). Κρατύλος / *Cratylus*. Consulted edition: L. Méridier (ed.), *Platon: Œuvres complètes*, tome V, 2e partie. Collection des universités de France. Paris: Les Belles Lettres, 1925.

Plato (*Prot.*). Πρωταγόρας / *Protagoras*. Consulted edition: Alfred Croiset & Louis Bodin (eds), *Platon: Œuvres complètes*, tome III, 1re partie, Collection des universités de France. Paris: Les Belles Lettres, 1925.

Plato (*Soph.*). Σοφιστής / *Sophista*. Consulted edition: Auguste Diès (ed.), *Platon: Œuvres complètes*, tome VIII, 3e partie. Collection des universités de France. Paris: Les Belles Lettres, 1925.

Politzer Robert L. & Ramirez Arnulfo G. (1973). 'An error analysis of the spoken English of Mexican-American pupils in a bilingual school and a monolingual school'. *Language Learning*, vol. 23, nº 1, pp. 39–62.

Pravec Norma A. (2002). 'Survey of learner corpora'. *ICAME Journal*, vol. 26, pp. 81–114.

Prytz Ylva Berglund (2007). 'Why is it full of funny characters? Converting the BNC into XML'. *Studies in Variation, Contacts and Change in English*, vol. 1: Annotating Variation and Change. <http://www.helsinki.fi/varieng/journal/volumes/01/berglund/>.

Przepiórkowski Adam & Bański Piotr (2009). 'Which XML standards for multilevel corpus annotation?'. In: *Proceedings of the 4th Language & Technology Conference*, Poznań, pp. 245–250.

Quirk Randolph (1960). 'Towards a description of English usage'. *Transactions of the Philological Society*, vol. 59, nº 1, pp. 40–61.

Quirk Randolph & Svartvik Jan (1966). *Investigating Linguistic Acceptability*. Janua Linguarum: Studia Memoriæ Nicolai van Wijk Dedicata, series minor, vol. 54. The Hague: Mouton.

Rosenberg Andrew & Hirschberg Julia (2007). 'V-measure: A conditional entropy-based external cluster evaluation measure'. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, pp. 410–420.

Russell Robert C. (1918). 'Index'. Specification of letters patent, nº 1,261,167. Washington: United States Patent Office.

Sampson Geoffrey (1995). *English for the computer: The SUSANNE corpus and analytic scheme*. Oxford: Clarendon.

Sampson Geoffrey (2007). 'Grammar without grammaticality'. *Corpus Linguistics and Linguistic Theory*, vol. 3, nº 1, pp. 1–32.

Santorini Beatrice (1990). 'Part-of-speech tagging guidelines for the Penn Treebank project (3rd revision)'. Technical report, nº MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, United States.

Schiftner Barbara (2008). 'Learner corpora of English and German: What is their status quo and where are they headed?'. *Vienna English Working Papers*, vol. 17, nº 2, pp. 47–78.

Schneider John & Kamiya Takuki (2009). 'Efficient XML interchange (EXI) format 1.0: W3C candidate recommendation 08 December 2009'. <http://www.w3.org/TR/2009/CR-exi-20091208/>.

Scholfield Phil (1995). *Quantifying Language: A Researcher's and Teacher's Guide to Gathering Language Data and Reducing it to Figures*. Clevedon: Multilingual Matters.

Scott Mike & Tribble Christopher (2006). *Textual Patterns: Key Words and Corpus Analysis in Language Education*. Studies in Corpus Linguistics, vol. 22. Philadelphia, United States: Benjamins.

Shutova Ekaterina (2009). 'Sense-based interpretation of logical metonymy using a statistical method'. In: *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and 4th International Joint Conference on Natural Language Processing of the AFNLP, Proceedings of the Student Research Workshop*, Singapore, pp. 1–9.

Sinclair John (2004). 'Intuition and annotation: The discussion continues'. In: Karin Aijmer & Bengt Altenberg (eds), *Papters from the 23rd International Conference on English Language Research on Computerized Corpora, Göteborg 22–26 May 2002*. Language and Computers: Studies in Practical Linguistics, vol. 49. Amsterdam: Rodopi, pp. 39–59.

Sinclair John M. (1992). 'The automatic analysis of corpora'. In: Jan Svartvik (ed.), *Directions in Corpus Linguistics: Proceedings of Nobel Symposium 82, Stockholm, 4–8 August 1991*. Trends in Linguistics: Studies and Monographs, vol. 65. Berlin: Mouton de Gruyter.

Sjöbergh Jonas (2005). 'Chunking: An unsupervised method to find errors in text'. In: *Proceedings of the 15th Nordic Conference of Computational Linguistics*, Åminne (Joensuu), pp. 180–185.

Sjöbergh Jonas & Knutsson Ola (2005). 'Faking errors to avoid making errors'. In: *Proceedings of the Conference on Recent Advances in Natural Language Processing*, Borovets, Bulgaria, pp. 1–6.

Smith Noah A. & Eisner Jason (2005*a*). 'Contrastive Estimation: Training log-linear models on unlabeled data'. In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor, United States, pp. 354–362.

Smith Noah A. & Eisner Jason (2005*b*). 'Guiding unsupervised grammar induction using contrastive estimation'. In: *Proceedings of the IJCAI Workshop on Grammatical Inference Applications: Successes and Future Challenges*, Edinburgh, pp. 73–82.

Sorace Antonella & Keller Frank (2005). 'Gradience in linguistic data'. *Lingua*, vol. 115, nº 11, pp. 1497–1524.

Svartvik Jan & Quirk Randolph (1980). *A Corpus of English Conversation*. Lund: Glerup.

Tetreault Joel R. & Chodorow Martin (2008). 'The ups and downs of preposition error detection in ESL writing'. In: *Proceedings of the 22nd International Conference on Computational Linguistics*, Manchester, pp. 865–872.

Thompson Henry S. & McKelvie David (1997). 'Hyperlink semantics for standoff markup of read-only documents'. In: *Proceedings of SGML Europe '97*, Barcelona.

Vlachos Andreas, Korhonen Anna & Ghahramani Zoubin (2009). 'Unsupervised and constrained Dirichlet process mixture models for verb clustering'. In: *Proceedings of the EACL workshop on GEometrical Models of Natural Language Semantics*, Athens, pp. 74–82.

Vlugter Peter, Ham Edwin van der & Knott Alistair (2006). 'Error correction using utterance disambiguation techniques'. In: *Proceedings of the Australasian Language Technology Workshop*, Sydney, pp. 123–130.

Wagner Joachim, Foster Jennifer & Genabith Josef van (2007). 'A comparative evaluation of deep and shallow approaches to the automatic detection of common grammatical errors'. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, pp. 112–121.

Wagner Joachim, Foster Jennifer & Genabith Josef van (2009). 'Judging grammaticality: Experiments in sentence classification'. *CALICO Journal*, vol. 26, nº 3, pp. 474–490.

Warburton (1747). *The Works of Shakespear: In Eight Volumes*. London: Knapton *& al.*

West Alfred W. (1894). *The Elements of English Grammar*. Pitt Press Series. 2nd edn, Cambridge: CUP.

Wible David, Kuo Chin-Hwa, Chien Feng-yi, Liu Anne & Tsao Nai-Lung (2001). 'A Web-based EFL writing environment: Integrating information for learners, teachers and researchers'. *Computers & Education*, vol. 37, nº 3–4, pp. 297–315.

Zephaniah Benjamin (1992). *City Psalms*. Newcastle: Bloodaxe.