Technical Report

UCAM-CL-TR-886
ISSN 1476-2986

Number 886

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Error detection in content word combinations

Ekaterina Kochmar

May 2016

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 December 2014 by the author for the degree of Doctor of Philosophy to the University of Cambridge, St. John's 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

This thesis addresses the task of error detection in the choice of content words focusing on adjective–noun and verb–object combinations. We show that error detection in content words is an under-explored area in research on learner language since (i) most previous approaches to error detection and correction have focused on other error types, and (ii) the approaches that have previously addressed errors in content words have not performed error detection proper. We show why this task is challenging for the existing algorithms and propose a novel approach to error detection in content words.

We note that since content words express meaning, an error detection algorithm should take the semantic properties of the words into account. We use a compositional distributional semantic framework in which we represent content words using their distributions in native English, while the meaning of the combinations is represented using models of compositional semantics. We present a number of measures that describe different properties of the modelled representations and can reliably distinguish between the representations of the correct and incorrect content word combinations. Finally, we cast the task of error detection as a binary classification problem and implement a machine learning classifier that uses the output of the semantic measures as features.

The results of our experiments confirm that an error detection algorithm that uses semantically motivated features achieves good accuracy and precision and outperforms the state-of-the-art approaches. We conclude that the features derived from the semantic representations encode important properties of the combinations that help distinguish the correct combinations from the incorrect ones.

The approach presented in this work can naturally be extended to other types of content word combinations. Future research should also investigate how the error correction component for content word combinations could be implemented.

Acknowledgements

First and foremost, I would like to express my profound gratitude to my supervisor, Ted Briscoe, for his constant support and encouragement throughout the course of my research. This work would not have been possible without his invaluable guidance and advice.

I am immensely grateful to my examiners, Ann Copestake and Stephen Pulman, for providing their advice and constructive feedback on the final version of the dissertation. I am also thankful to my colleagues at the Natural Language and Information Processing research group for the insightful and inspiring discussions over these years. In particular, I would like to express my gratitude to Øistein Andersen, Helen Yannakoudakis, Andrew Caines, Tamara Polajnar, Aurelie Herbelot, Eva Vecchi, Marek Rei, Mariano Felice and Zheng Yuan. I would like to thank Paula Buttery and Stephen Clark for giving early feedback on my work and helping to shape this research at the early stages. Special acknowledgement goes to Diane Nicholls for annotating the learner data and offering her expertise and advice, and to Øistein Andersen for his invaluable help in learner data annotation, among other things, and for providing his guidance.

I would like to acknowledge Cambridge Assessment and Cambridge University Press for granting permission to use learner data and making this research possible. My studies have been funded by the Cambridge Trusts and Cambridge Assessment, and I am extremely grateful to them for giving me the opportunity to conduct my research in Cambridge. I am also very thankful to the Computer Laboratory, St. John's College and Cambridge Assessment for supporting my conference attendance.

Finally, my profound thanks to my family and friends for their faith in me, their incessant love and support.

Contents

List of abbreviations

Acc	Accuracy, *p. 67*
add	Additive model, *p. 58*
alm	Adjective-specific linear maps, *p. 58*
AN	Adjective–noun combination, *p. 26*
BNC	British National Corpus, *p. 34*
CEFR	Common European Framework of Reference for Languages, *p. 88*
CLC	Cambridge Learner Corpus, *p. 14*
CLC-FCE	Cambridge Learner Corpus First Certificate in English, *p. 27*
CLEC	Chinese Learner English Corpus, *p. 33*
DSM	Distributional semantic models, *p. 54*
DT	Decision Tree, *p. 132*
EA	Error analysis, *p. 29*
ED	Error detection, *p. 13*
EDC	Error detection and correction, *p. 13*
EFL	English as a Foreign Language, *p. 14*
ESL	English as a Second Language, *p. 14*
ESOL	English as a Second or Other Language, *p. 14*
F	F-measure, *p. 68*
FN	False negatives, *p. 66*
FP	False positives, *p. 66*
HOO	Helping Our Own, *p. 13*
IC	In-context, *p. 79*

ICLE	International Corpus of Learner English, *p. 33*
IELTS	International English Language Test System, *p. 13*
L1	First language, *p. 14*
L2	Second language, *p. 14*
LM	Language modelling, *p. 41*
LMI	Local Mutual Information, *p. 64*
ML	Machine learning, *p. 26*
MRR	Mean reciprocal rank, *p. 48*
mult	Multiplicative model, *p. 58*
NPMI	Normalised pointwise mutual information, *p. 91*
NUCLE	National University of Singapore Corpus of Learner English, *p. 32*
OOC	Out-of-context, *p. 79*
P	Precision, *p. 67*
PLSR	Partial least squares regression, *p. 64*
R	Recall, *p. 67*
SAT	Self-assessment and tutoring, *p. 22*
SLA	Second language acquisition, *p. 13*
SMT	Statistical Machine Translation, *p. 41*
SVD	Singular Value Decomposition, *p. 64*
TN	True negatives, *p. 66*
TOEFL	Test of English as a Foreign Language, *p. 25*
TP	True positives, *p. 66*
VO	Verb–object combination, *p. 26*
*	is used to indicate error
?	is used to indicate linguistic strangeness

Chapter 1

Introduction

This thesis presents work on error detection in content word combinations in non-native writing of English. This research is related to a number of topics ranging from *second language acquisition* (*SLA*), to *learner language*, to *learner data annotation*, to automated *error detection* (*ED*) and *error detection and correction* (*EDC*). In this chapter, we give a brief overview of the areas involved in this research.

Research on SLA and learner language has a long history. People have always been interested in speaking other languages in order to communicate with each other, and English has become a widely used *lingua franca* for people from different countries not sharing a common native language. In addition, in certain spheres of life a good command of English is a vital requirement. For example, the number of students entering educational programs in foreign universities is growing every year. Universities offering international programs run courses in English, and in order to enter such courses students are required to pass language tests such as the *International English Language Test System* (*IELTS*) administered by *Cambridge English Language Assessment* to demonstrate adequate competence in English. Science and research is another sphere in which good command of English is of great importance as researchers should be able to convey and exchange their ideas with their colleagues from all over the world. This issue has been a focus of two shared tasks on EDC – *Helping Our Own* (*HOO*) 2011 and 2012 – which aimed at promoting the use of *Natural Language Processing* (*NLP*) tools and techniques to help improve the textual quality of academic papers on NLP written by non-native speakers in the field (Dale and Kilgarriff, 2011; Dale et al., 2012).

English is a non-native language for the majority of people who use it: according to Crystal (2003, p. 69), non-native speakers of English outnumber native speakers by a ratio of three to one, and according to Chodorow et al. (2010), they are a large and growing section of the world's population. It is estimated that in China alone 300 million people are currently studying English. Guo and Beckett (2007) report that over a billion people speak English as a second or further language. Tetreault and Chodorow (2008a) report that even in predominantly English-speaking countries, the proportion of non-native speakers can be substantial, and in 2002, the US National Center for Educational Statistics reported that nearly 10% of students in the US public school population speak a language other than

English and have limited English proficiency. It is conventional to distinguish between *English as a Second Language* (*ESL*) and *English as a Foreign Language* (*EFL*) with the former being used for English learned in places where it is spoken on a par with one or more other languages and where learners have constant exposure to it, and the latter being related to countries in which English is not spoken. In this research, we do not distinguish between ESL and EFL and focus on non-native English in general, or on *English as a Second or Other Language* (*ESOL*).

Due to the diversity of languages in the world, the learning of English proceeds at various rates. Research on SLA has addressed a wide variety of questions related to language learning such as how difficult it is for native speakers of different languages to learn English, how much their progress in learning relies on their native language (or *L1*), whether the errors they commit could provide valuable information and guide further learning, and to what extent these errors can be predicted. Answers to these questions could lead to better language teaching strategies and help improve learners' English. We review previous SLA research in §1.1.

The growing number of non-native speakers of English, the wide-ranging difficulty of learning and the demand for good language skills create a constantly increasing need for tools to support English language learning and instruction at all levels and in all countries (Chodorow et al., 2010). In addition to the tools for language learning such as self-tutoring systems, there is a constant demand for tools that can improve one's writing by automatically detecting and correcting errors. Not surprisingly, automated EDC is a developing field of research. We review this field in §1.2.

Most previous research in EDC has focused on function words (Leacock et al., 2010, 2014). Function words, being some of the most frequently used elements of language as well as some of the most difficult to master in a foreign language (or *L2*), cover a substantial portion of learner errors. Errors in the choice of content words, such as nouns, verbs, adjectives and adverbs, also constitute a substantial part of learner errors – they are the third most frequent error type after errors in determiners and prepositions (Leacock et al., 2010). Moreover, content words convey essential elements of meaning, so the appropriate choice of content words is crucial for successful writing. Yet to date EDC for content words is mostly under-explored and offers much room for improvement. In this thesis, we focus on errors in content word combinations and discuss the specific challenges and goals for EDC of content words. We show that the errors in the use of content words and function words have to be tackled differently.

The availability of learner corpora and texts produced by language learners is central to any research on EDC. In this project, we use the *Cambridge Learner Corpus* (*CLC*) which is a collection of texts written by non-native speakers in response to prompts as part of Cambridge ESOL examinations. These essays represent freely generated text only restricted by the topic of the essays, and this data is valuable for error-related research. As there has not been much research on content word combination errors, one of the important steps of this project has been the collection and thorough annotation of a dataset of learner errors in content word combinations. This dataset has been extracted from the CLC and represents real-life learner errors. The annotations provide information about specific error types in content word combinations. §1.3 summarises the main goals and motivation for the research presented in this thesis.

1.1 Second Language Acquisition

Second Language Acquisition focuses on how English as an L2 is acquired and what the underlying mechanisms of language learning are. The communicative goal can be seen as the ultimate goal of language learning: it is important for learners to express themselves clearly in the L2 so as to be understood by native speakers of English as well as by speakers with other L1s. Hence, learners might aim to speak and write in a World Standard English, or a norm of English. Definitions of a **norm** and an **error** are central to EDC research.

Corder (1971) suggests not considering the text produced by learners 'erroneous' as the term 'error' implies wilful breach of rules that are supposed to be known, whereas no wilful breach can occur when learners do not know the relevant rules of the L2. As Corder points out, what appears to be deviant in comparison to the L2 might be correct within the learner's idiosyncratic dialect (or *interlanguage* as per Selinker (1972, 1992)) and conform to the rules the learner knows at a certain point of their L2 acquisition process. However, if we adopt the communicative goal as the ultimate goal of language learning, we see that it is the rules of the L2 that the learner should conform to in order to be understood by a wide community of speakers with different L1s. Therefore, the definition of learner error should address the discrepancies between what learners produce and what native speakers of the target L2 produce. James defines an **error** as "an unsuccessful bit of language" (James, 1998, p. 1) or as "an instance of language that is unintentionally deviant and not self-corrigible by its author" (James, 1998, p. 78). Frei (1929) suggests distinguishing between rules of grammar and rules or laws of society, as certain language conventions are not clearly defined. As we shall see later, certain errors including some content word errors are explained by the breach of such language conventions rather than strictly defined rules.

The crucial step in defining a learner error is setting the **norm** of English. A norm stabilises language around one accepted variant of language making it more efficient and less variable, thus facilitating communication. Andreasson (1994) argues that without a norm, language, being based on a set of arbitrary conventions, would break down. James (1998, p. 35) advocates the use of the 'official' or national norm, referred to as *standard English*, which should be established as a pedagogic norm of native-speaker English and should be taught at schools. The norm is contrasted with other varieties of native-speaker language – non-standard dialects, or non-standard 'home' languages – and students' performance should be measured against the accepted norm. The definition of a norm is more important for written than for spoken English, since spoken language is often more variable: speech allows for interaction between speakers, while the purpose of a written text is usually to transfer the message to the reader in a non-interactive way.

James (1998, p. 39) also notes that there are different varieties of standard English spoken around the world, including the older Englishes (British, American, Canadian, Australian, and New Zealand) which are the original norm-providers for those who learn English as an L2. In this work, we use British English as the norm.

The definition of a norm is, thus, based on the language used by native speakers. Native speakers are assumed to be perfect in their mastery of language and are allowed, unlike non-native speakers, to make up their own rules: "We believe, as most linguists, that native speakers do not make mistakes. Native speakers for the most part speak their native language perfectly" (Andersson and Trudgill, 1990, p. 111). However, text produced by native speakers is not always error-free, and it is common for them to make mistakes, use non-standard language or innovate. Native speakers constantly introduce novel forms and words, some of which then become rooted in the language and are eventually considered part of the norm. Non-standard use is common for figurative language, poetry, jargon, and similar: for example, *fun-size Chokko bars* used in advertising, or Dylan Thomas' "*Once below a time...*". Since native speakers are considered to be experts in their own language, whenever they use borderline word constructions, it is attributed to reasons other than mere incompetence: for example, to creativity in using language.

One distinction between native and non-native speakers is the amount of authority assigned to them. Mey (1981) suggests that there is a natural scale reflecting the native speakers' authority: they are at their most authoritative on matters of phonology, less so on morphology, less still on syntax, and least on semantics. Semantics occupies the extreme end of this scale as it is the most diverse and changing area in linguistics with new meanings and new words being created all the time. The same scale reflects native speakers' tolerance of linguistic deviance: they are least tolerant to phonological deviance and most tolerant to semantic deviance. Phonetic deviance is almost exclusively interpreted as mere incompetence with little creativity allowed in this area. James (1998, p. 146) also points out that the claim that native speakers know their own language perfectly might be true for some linguistic levels (e.g., syntax), but not necessarily for others (e.g., lexis). Native speakers can be ignorant of some of the lexical stock of their own language, and they continue to accumulate lexical knowledge throughout their lives.

James (1998, p. 64) distinguishes between several categories of learners' ignorance, and in particular distinguishes between *grammaticality* and *acceptability*:

- ***Grammaticality*** or well-formedness has to be judged on objective grounds and on the basis of violation of clearly defined rules of language use. If for some piece of language one can say that there are no circumstances in which it could be formulated in this way, the construction is ungrammatical: for example, *he *jump* or *important *informations*. This category is context-independent. James (1998, p. 66) concludes that some semantic and collocational anomalies like *The milk turned *rotten* or *A *flock of elephants* do not belong to this category as the conclusions about such word combinations are not within the sphere of grammar, but rather have to do with the speaker's intentions and hearer's judgements. It depends on whether the speaker meant *rotten* or *sour*, *flock* or *herd* as, for example, might be the case if the words were used metaphorically. James (1998, p. 66) concludes that in such cases we are dealing with the user's viewpoint, or with ***acceptability***. This view is similar to that of Chomsky who noted that "the notion 'grammatical' cannot be identified with 'meaningful' or 'significant' in any semantic sense" and that "any search for a semantically based definition of 'grammaticalness' will be futile" (Chomsky, 1957, p. 15).

- ***Acceptability*** has to do with non-linguistic factors, but grammaticality is seen as a prerequisite for acceptability: "An acceptable utterance is the one that has been, or might be, produced by a native speaker in some appropriate context and is, or would be, accepted by other native speakers as belonging to the language in question” (Lyons, 1968, p. 137), and “To understand a sentence, then, it is first necessary to reconstruct its analysis on each linguistic level; and we can test the adequacy of a given set of abstract linguistic levels by asking whether or not grammars formulated in terms of these levels enable us to provide a satisfactory analysis of the notion of ‘understanding’” (Chomsky, 1957, p. 87). De facto use and unproblematicity are the tests for establishing acceptability: to decide whether an utterance is acceptable or not, one should try to think of the context where it can be used appropriately. James (1998) offers the following example: suppose a native speaker is asked to judge the acceptability of the utterance *Pele (the Brazilian footballer) wore a green dress.* If a native speaker is asked to judge whether this is grammatically correct, they might reply “Yes, if he were taking part in the Rio carnival celebrations”. A particular context of use might reveal that the learner meant ‘shirt’ rather than ‘dress’ as they were talking about the player’s outfit during a football match.

To decide whether something is acceptable, even when it satisfies the grammaticality test, is seldom clear-cut and takes some thought or even imagination. Besides ungrammaticality and failure to fit the intended context as in *Pele wore a green ?dress and ?made three goals*, a phrase may be judged unacceptable if it, for example, expresses an unconceivable idea (as in *My lawnmower ?thinks that I don’t like it*), contains flouting customary collocations (as in *The ?white and black cat grinned like a ?Cornishman*), uses unusual grammar or phonological configurations (as in *He was finishing doing computing approaching retiring*), or is hard to process syntactically or phonologically (as in *The flea the rat the cat the dog chased killed carried bit me*). Acceptability can also be equated with processibility – an ability of the listener or reader to extract the meaning of the message transferred.

Different error types can be attributed to different categories in this scheme. Most grammatical errors and errors in the use of articles are caused by violations of *grammaticality* as they breach clear-cut rules of language. Errors in content words are related to *ungrammaticality* if there is no context where a given word combination can be used appropriately, and *unacceptability* if the combination in question does not fit the intended context.

The incompleteness hypothesis states that full mastery of an L2 might not be attainable for non-native speakers. Even when a text is correct, learners with different L1s can produce the same text based on different assumptions from their interlanguage. One of our goals is to identify, as far as possible, the reasons for the content word selection errors committed by language learners. However, we recognise that learner language might never achieve nativeness and in that respect we follow some researchers who advocate that the goal of ESOL learning should be a competent rather than native-like knowledge of English. For example, Chomsky (1986, p. 16) advocates a ‘scientific’ approach to describing the learner’s English that would “say that the person has a perfect knowledge of some language L, similar to English but still different from it”, and Cook (1991, p. 114) states that “The model for language teaching should be the fluent L2 user, not the native speaker”. Our primary goal is to identify clearly deviant uses of content words that impede understanding and that would be identified by native speakers as unacceptable.

1.2 EDC in learner data: challenges and benefits

1.2.1 General principles

James (1998, p. 91) distinguishes between the following steps in error **detection**:

- ***error detection proper*** which is concerned with identifying whether something is an error or not;
- ***error location*** which is concerned with identifying the exact location and span of an error;
- ***error description*** or the choice of a meta-language for describing an error with the purpose of explaining, preventing and repairing errors;
- ***error classification*** or ***categorisation*** aimed at categorising and counting errors by type that can further be used to investigate ED for type-specific errors and to alert learners to the type-specific problems;
- ***error diagnosis*** which attempts to explain potential reasons for the errors committed: for example, *interlingual* or *intralingual*.

Each of these steps presents different challenges. James (1998, p. 91) points out that it is easier for people to spot errors in written rather than spoken language, and also in somebody else's writing rather than in one's own. For that reason, error annotation is usually performed by human experts in the field: by teachers of English in a classroom environment, or by trained experts. The performance of an automated ED system is then compared to human judgements on the same task, as is the usual practice in NLP. It is unrealistic to expect that an automated system would perform better than humans, therefore the inter-annotator agreement on ED and EDC can be used as an informative upper bound for an automated system.

Studies show that neither *error detection* nor *error location* are straightforward tasks for humans. Agreement between human annotators varies depending on the error type: it is easier to agree on certain grammatical errors, but errors related to discourse and semantics may be harder to agree upon. This is related to the different level of tolerance towards different types of errors (see §1.1): the more variability is allowed in a certain linguistic domain, the less clear-cut are the rules defining the correct usage. The natural difficulty of detecting the span and type of an error can be illustrated by overlapping and interacting errors: for example, **Book inspire me* allows for two competing corrections – a missing article error and a subject-verb number agreement error with the correction *The book inspires me*, or a subject-verb number agreement error with the correction *Books inspire me* (Leacock et al., 2014, p. 35). In some, but not all, cases the surrounding context may help choose one correction over the other.

Error taxonomies used for *error categorisation* can be built using different principles: they can be based on linguistic categories (e.g., parts of speech), or on surface structure. The latter can include the following categories:

- *omission* of a word or phrase;
- *addition* of unnecessary bits of text;
- *misformation* or *misselection* which corresponds to a wrong word choice and includes, among others, *archiform* errors discussed below;
- *misordering* which corresponds to the wrong word order and often results from word-for-word translation from one's L1; and
- *blends* which result from contamination of more than one well-defined target in the learner's mind.

Omission is more typical for function words, whereas when it comes to the use of content words, learners tend not to omit words but rather paraphrase. Addition manifests itself differently in different linguistic categories: addition with respect to determiners means using an unnecessary determiner when none is needed, whereas with respect to content words it covers verbosity in an attempt to describe something for which a learner has no suitable word in their mental lexicon as in *things that come every week on TV* for *TV serials*.

Dulay et al. (1982, p. 160) discuss misselection errors and, in particular, talk about archiform errors which they define as "selection" by the learners "of one member of a class of forms to represent others in the class". This occurs in both function and content words. For example, learners tend to choose *that* to represent the class [*this/that/those/these*]. As a result, *that* tends to be overrepresented in learner language, while the other members are underrepresented. A number of errors in the choice of content words can also be explained by misselection of archiform as we shall see in §2.3.2: it is common for learners to choose a word with a more general meaning, for example *big*, to represent a whole class of words related to size, and inappropriately use it instead of more specific terms like *broad*, *long* or *wide*.

Blends describe the phenomenon of error hybridization when a learner says, for example, **according to Erica's opinion* instead of either *according to Erica* or *in Erica's opinion*, or when *a typical Indian meal comprises rice, dhal...* is blended with *a typical Indian meal is comprised of...* and results in *a typical Indian meal *comprises of...* (James, 1998, p. 112). This error can be viewed as a blend of the two structures, and could also be caused by similarity to the semantically close verb *composed*. Semantic issues play an important role in such errors, but identifying a particular source for the error in such cases is a difficult task for human annotators.

We primarily focus on misselection as it is the most frequent error category in content words.

With respect to error **correction**, James (1998, p. 236) distinguishes between the following types:

- *feedback* which aims to inform learners that there is an error, and leave them to discover and repair it themselves;

- *correction proper* which aims to indicate that the present attempt is wrong, specifying how and where it is wrong, and suggest an alternative or give a hint;

- *remediation* which aims to provide learners with information that allows them to revise or reject the wrong rule they were operating with when they produced the error token.

Similar feedback type taxonomies have been used elsewhere in literature with some difference in terminology: what James (1998, p. 236) refers to as *feedback* is often called *indirect feedback*, while *correction proper* is also referred to as *direct corrective feedback*, and *remediation* as *direct corrective feedback with additional meta-linguistic information.* Each of these types serves a particular purpose, and the choice of the feedback type to provide with an EDC system will depend on the goals as well as the complexity of the algorithms used. It can be argued that *feedback* in James' terminology is the most general of the three types, while *remediation* is arguably the most difficult and costly to provide automatically. *Feedback* and *correction proper* usually serve a short-term goal of detecting and correcting errors in a particular piece of writing and are realised by teachers correcting students' writing as well as by automated EDC systems and spell- and grammar-checkers. *Remediation* is oriented more to the long-term effect, and is realised by teachers providing further explanations and automated tutoring systems which aim to detect the source and reason for an error and provide instructions on how to prevent such errors in the future. Depending on the type of error committed and on the learner's level of English, some learners might benefit more from *feedback* while others might need *correction* and *remediation.*

1.2.2 Usefulness of corrective feedback

The question that has to be answered within EDC is whether learners benefit from the feedback provided, whether it improves their writing and facilitates language learning, in particular when the feedback is provided by an EDC system. For automated feedback to be useful, it should be accurate enough, as well as clear and informative for learners to understand and attend to the system's suggestions.

The relevant issues for corrective feedback on learner writing include:

- ***goal***: *short-term* aimed to improve the quality of a given piece of writing, or *long-term* aimed to facilitate language learning;

- ***focus***: *focused* on one particular error type, or *unfocused* covering several error types;

- ***type***: *direct* comprising both detection and correction, or *indirect* consisting of an indication that there is an error without further specification;

- ***source of feedback***: whether it is provided by a teacher or by a tutoring or EDC system;

- ***amount of supporting information***: learners can additionally be provided with explanatory examples scraped from the Web, as was implemented in the *ESL Assistant* (Chodorow et al., 2010), or pointed to the relevant chapters in a grammar book as in *Criterion* (Lipnevich and Smith, 2008). Alternatively, they can be provided with a short explanation or an oral session by a teacher.

EDC systems can be evaluated in a *system-centric* or in a *user-centric* manner (Chodorow et al., 2010). The former focuses on how well the system detects and corrects the errors that it is supposed to detect and correct, and it is this aspect of EDC systems that is most commonly examined. The latter is also important as it focuses on how the system impacts the quality of writing and whether learners actually benefit from the feedback.

An early study by Truscott (1996) claimed that grammatical error correction in L2 writing is actually ineffective and harmful for language learning. This motivated a number of researchers to question whether corrective feedback is helpful for learners, and if so, which type of feedback is most helpful. Some of these studies looked into feedback provided by teachers (Bitchener, 2005; Sheen, 2007; Bitchener et al., 2008; Ellis et al., 2008). A number of studies in recent years looked into the usefulness of computerised feedback and evaluated the systems in a user-centric manner (Attali, 2004; Lipnevich and Smith, 2008; Chodorow et al., 2010; Nagata and Nakatani, 2010; Andersen et al., 2013).

The key findings of these studies can be summarised as follows:

- All studies confirmed a positive effect of corrective feedback: a significant improvement in accuracy was retained between the immediate and delayed post-tests. In contrast, the control group which received no feedback showed either a decline in accuracy, or inconsistent performance (Bitchener et al., 2008; Ellis et al., 2008), or was able to only show an insignificant improvement over time which might be attributed to a self-learning effect when learners pass multiple writing tests of a similar kind (Sheen, 2007). This refutes the original claim by Truscott (1996) concerning the ineffectiveness of corrective feedback.

- The studies that included meta-linguistic information and compared direct feedback alone to direct feedback with additional meta-linguistic explanation showed that the latter results in stronger effects over time (Sheen, 2007). It can be argued that whereas both types of feedback are likely to promote learners' awareness as *noticing*, only direct corrective feedback with meta-linguistic comments promotes awareness with *understanding* and facilitates learning.

- Some studies assumed that focused feedback for a single or limited number of grammatical issues can be more beneficial for language learners who have limited processing capacity and are not able to deal with information overload when presented with unfocused corrective feedback (Sheen, 2007). The results of Ellis et al. (2008) who compared the performance of learners who received focused feedback on articles with those who received unfocused feedback on multiple error types show that there is a substantial improvement in accuracy in both groups, while the differences in improvement between the groups are not statistically significant. The accuracy is maintained over time in the unfocused feedback group and improves in the focused feedback group, which suggests that in the long run focused feedback might, indeed, be more informative and useful.

The results of these studies can be interpreted as a proof that high quality corrective feedback facilitates language learning and results in improved accuracy in the use of certain linguistic categories, with this positive effect being retained over time. When it comes to implementation of an EDC system, an additional challenge arises: the feedback provided by the system should be of high quality, comparable to that provided by a teacher. Meta-linguistic explanation is also harder to generate automatically.

The usefulness of the diagnostic feedback provided by three automated systems has been shown so far: Attali (2004), Lipnevich and Smith (2008) and Chodorow et al. (2010) discussed the usefulness of the feedback provided by *Criterion*^{SM} developed by Educational Testing Service; Chodorow et al. (2010) also showed the usefulness of the *ESL Assistant* developed by Microsoft Research; and Andersen et al. (2013) presented and discussed a *Self-Assessment and Tutoring* (*SAT*) system developed by the University of Cambridge and aimed at intermediate learners of English.

The usefulness of an EDC system can be proved by a decreased error rate in the writing of learners who use the system, while negative findings might suggest that learners have difficulties in understanding the automatically generated feedback or the suggested corrections. All studies confirmed an improvement in learner writing resulting from the use of these automated systems.

The systems use different approaches to error correction: for example, *Criterion* uses meta-linguistic feedback, labelling the type of error and pointing the learner to the relevant chapter in a grammar book. *ESL Assistant* incorporates web search for both the original string and the suggested correction if the system identifies the original use as an error. This type of feedback mimics non-native speakers' behaviour when they use web search in order to verify correct usage of English expressions, and has the additional advantage of allowing learners to make the final decision about the correct linguistic form. Chodorow et al. (2010) investigated how often learners found examples useful, how often they accepted the suggestions and whether the acceptance was informed rather than blind. The study confirmed that the learners indeed make selective decisions, and moreover, can distinguish between valid and invalid system suggestions. While confirming that ED can be automated, this study also showed that the correction step can be implemented as an interactive process, especially when the errors are related to semantics. The *SAT* system (Andersen et al., 2013) provides automated feedback of three types and at different levels of granularity: an holistic score reflects learners' proficiency and represents evaluative feedback, a score for each sentence highlighting well-written as well as less well-written passages represents indirect corrective feedback, while specific comments on local issues including spelling and word choice represent direct corrective feedback. At the sentence level, learners are made aware of the problematic areas but have to figure out the corrections themselves, while at the level of individual errors they are provided with suggestions rather than prescriptive corrections, so the system involves a great deal of interactivity and makes the learners think and analyse rather than automatically accept the system's suggestions. A user questionnaire revealed that learners found this approach useful, facilitating their critical thinking and analytical skills.

It should also be noted that all three systems aim at high precision in ED (Chodorow et al., 2010; Andersen et al., 2013): Chodorow et al. (2010) reported precision of about 90% for article ED and 80% for preposition ED for *Criterion*, while the *ESL Assistant* achieved 91% precision for article ED and 78% precision for preposition ED. It has been noted before (Leacock et al., 2014) that false positives, or correct instances misidentified by an ED system as errors, are notoriously annoying for users. It could be argued that false positives are also highly misleading for language learners, and this could result in a negative effect on learning. Nagata and Nakatani (2010) argue that when only a limited number of errors is detected with high precision, learners can detect the other incorrect instances by generalising the system's feedback to more instances of the same kind using their knowledge of English, and such activities facilitate language learning. In contrast, if a system detects errors with only limited precision, learners focus on judging whether the given results are trustworthy or not, and do not learn much from such feedback. Their findings confirm that imprecise feedback misleads learners more than no feedback.

The studies on usefulness of automated corrective feedback have so far mostly focused on certain error types and there is no conclusive evidence about what type of feedback is most useful for error categories involving content words. However, it is possible to draw some general conclusions:

- Automated corrective feedback helps improve accuracy of learners' writing provided that it is accurate and precise. The studies report precision above 0.80.
- An EDC system should aim for high precision as imprecise feedback misleads learners and has a negative effect on their learning progress.
- Learners are able to attend to the systems' comments and make informed choices when presented with possible corrections and additional information on those. Error correction, especially for errors related to semantics such as the ones in the choice of content words, can be implemented as an interactive process, while error detection should be automated.
- Learners are able to use indirect automated feedback and report that they find it useful.

We aim to implement an ED system with the focus on errors in content word combinations. ED can be seen as serving the short-term goal of detecting errors in a particular piece of writing; however, if an automated system detects errors with high precision it also serves the long-term goal of language learning.

1.2.3 ED in content words

Tools for grammatical analysis and correction in written text have been around for several decades. At the earlier stages of research on EDC, grammar checking tools were based on string matching and involved little, if any, linguistic analysis (for example, the *Unix Writer's Workbench* (Macdonald et al., 1982)). Later systems, such as *Correct-Text* (Houghton Mifflin Company) and *Grammatik* (Aspen Software) performed some linguistic analysis, while some other tools like IBM's *Epistle* (Heidorn et al., 1982) and *Critique* (Richardson and Braden-Harder, 1988) ran full linguistic analysis using sophisticated grammars and parsers (Leacock et al., 2014, p. 7). The use of grammars and parsers allowed these systems to target a wide range of grammatical errors; still the algorithms relied heavily on hand-coded rules. As of today, some grammar checking tools still involve some rule-based heuristics. Rule-based approaches to grammatical analysis are efficient for the types of learner errors that can be described and corrected using a clearly defined set of rules, for example subject-verb agreement. For instance, *ESL Assistant* used a combination of rule-based and statistical machine-learning approaches: the former were applied to the error types amenable to simpler solutions, such as noun number, verb formation or irregular verb errors, and the latter were used for the error types, such as article and preposition errors, that are difficult to identify and resolve without taking complex contextual interactions into account (Leacock et al., 2009).

With the emergence of large-scale annotated treebanks and other resources, as well as statistical parsers trained on these resources, statistical approaches to grammatical analysis as well as to EDC in learner data began to dominate the field. Statistical systems assign high probabilities to the sequences of words that are seen or are substantially similar to the sequences seen during training, while unobserved sequences receive low probabilities and have a higher chance of being identified as errors. This approach is efficient for EDC in cases when usage is reliably covered by the data. For example, idiomatic expressions can be expected to have reliably high counts in native data. As Nunberg et al. (1994) point out, idioms are often considered to "typically appear only in a limited number of syntactic frames or constructions, unlike freely composed expressions" (Nunberg et al., 1994, p. 492). This property has been shown to not always hold for idioms (Nunberg et al., 1994; Riehemann, 2001). However, Riehemann (2001, p. 32) still points out that "for each idiom there is a particular fixed phrase (modulo inflection of the head) which is recognized by speakers of the language as the normal form this idiom takes, and which is used much more frequently than would be predicted from independent factors". Riehemann (2001, Chapter 3) shows with a number of examples that the canonical forms of the idioms are used in the native data much more frequently than modified forms. Extending the argument about the native speakers' authority from §1.1, we can also assume that while native speakers can use modified idioms, language learners will still be expected (or even encouraged) to use the idioms in their canonical rather than modified form. Therefore, statistical approaches can help identify that *hit the bucket* used by a language learner is potentially a failed idiom with a possible correction being *kick the bucket*. The key problem with statistical approaches aimed at replicating observed combinations is that human language is highly productive, especially when it comes to the use of content words in freely composed expressions, and no corpus can effectively sample all possible content word combinations (see the discussion in Chomsky (1957, p. 15)).

The increasing availability of learner corpora in recent years has allowed the extraction of information about the typical errors committed by language learners of different L1 backgrounds at various stages of their language proficiency. This information has various applications in EDC: from training statistical classifiers on incorrect rather than correct uses (Han et al., 2010; Dahlmeier and Ng, 2011b), to using typical error confusion patterns and adapting machine learning classifiers to L1-specific priors (Rozovskaya and Roth, 2011), to developing systems that can identify with high precision frequently occurring errors in learner writing (Kochmar et al., 2012; Andersen et al., 2013).

The field of EDC in recent years has mainly focused on function words such as determiners and prepositions since they are notoriously difficult for language learners to master and as such represent a large subset of learner errors. Certain properties of these linguistic elements make it possible to treat EDC of function words as a classification problem and detect and correct such errors using machine learning classifiers. For instance, since these words belong to closed classes, the likely error patterns represent a small finite set. A classifier can be trained on well-formed text to learn a model of the correct use for an article or preposition. The features for the classifier are extracted from the surrounding context which is usually highly informative for function words. At application time, the classifier compares the class returned based on information learned from the training data with the word (class) originally used by the learner, and detects an error if the classes do not coincide. This approach proves to be efficient for function words due to the limited number of the classes representing closed-class words such as articles and prepositions, and due to the fact that the surrounding context can provide information about correct usage and generalise to new data. The same is not applicable to errors in content words, which cannot be described in terms of a finite set of possible confusions, while the correct use of the content word itself is defined by language conventions and semantics rather than syntax or grammatical relations between words.

Leacock et al. (2010, 2014) note that errors in content words and collocations represent a substantial portion of learner errors. According to them, these errors are the most common after the incorrect use of articles and prepositions.[1] In addition, errors in content words are also potentially more harmful as they change the intended meaning of the text and impede understanding. Leacock et al. (2014, p. 21) mention an experiment in which teachers of English were asked to rank errors according to their gravity. "The teachers had high agreement, ranking the two most serious errors as being word choice (*sky* versus *heaven*) and preposition errors". Leacock and Chodorow (2003) built a regression model to match the error types with holistic scores in the *Test of English as a Foreign Language* (*TOEFL*). Errors resulting from confusion between homophones and confusion of morphologically related forms were among the top five most useful predictors of the score in their study, which confirms that errors in content words have a direct impact on overall results.

In spite of being very frequent in learner writing and impeding understanding, errors in the choice of content words have received much less attention and to date remain an under-explored topic in the field of EDC. Certain properties of the use of content words make these error types very challenging for EDC algorithms. We aim to fill this gap in EDC research and explore methods of detecting and correcting errors in content words.

1.3 Project goals

1. The current work focuses on ED in the choice of content words. We address errors

[1] Precise figures depend on errors covered by the term "collocation errors". For example, Dahlmeier et al. (2013) list the "wrong collocation" error type among the five most frequent error types in the NUCLE corpus, but they combine a wide variety of errors including the ones in the use of prepositions and particles under one error-type label.

in two particular types of content word combinations – *adjective–noun* (*AN*) and *verb–object* (*VO*) combinations, as they cover a substantial portion of learner errors in the use of content words. The approaches to ED presented in this work can naturally be extended to other types of content word combinations.

2. Content word combinations allow for higher variability and do not follow any clear-cut rules of English. Therefore, we argue that with respect to content words learners benefit most from error detection and should be notified of incorrectly chosen content words in their writing. Error correction, on the other hand, can be performed in an interactive way, and final decisions about corrections can be left to the learner. An ED algorithm fulfils a short-term goal, but provided that error detection is performed with high accuracy, it also facilitates language learning and should lead to long-term effects.

3. It has been shown that precision-oriented ED approaches facilitate language learning. In this research, we aim for high precision in error detection.

4. In view of the lack of error-annotated datasets on content word combination errors, we collect and annotate a dataset of AN and VO combinations. For that, we analyse learner errors in the texts produced by language learners in response to essay prompts, which means that the errors are extracted from freely-generated texts and exemplify typical learner errors. The annotation scheme is devised to cover, describe and explain the errors committed. The collected and annotated datasets are freely available and are useful from both theoretical and practical points of view: they can be used to investigate error categorisation in content word combinations as they represent typical errors committed by language learners, and they can also be used as a testbed for EDC algorithms applied to content words.

5. We argue that incorrect content word combinations often exhibit semantic mismatch between the words chosen, and that approaches based on semantics are suitable for detecting errors in content word combinations. We implement compositional distributional semantic models, and we show how they can be applied to the learner data to detect errors in the choice of content words. We use the output of these models and derive "semantically informed" features which we use with a *machine learning* (*ML*) classifier. We show that this semantically motivated approach outperforms the other approaches that currently represent state of the art.

1.4 Thesis structure

- Chapter 2 presents the theoretical background of the current work. We overview the related areas in EDC, and discuss learner corpora collection in §2.1, learner data annotation principles in §2.2, approaches to EDC in function words as well as in content words in §2.3. We motivate the usefulness of models of compositional distributional semantics for ED in content word combinations in §2.4, and finally discuss how the systems are evaluated in §2.5.

- Chapter 3 presents the datasets of learner errors in AN and VO combinations. In the experiments presented in this thesis we use datasets extracted from the publicly-available and error-annotated *Cambridge Learner Corpus First Certificate in English* (*CLC-FCE*) dataset, and we describe these datasets in §3.1. We also present new datasets of AN and VO combinations that are extracted from the full CLC and contain combinations that are not attested in a native corpus of English. We discuss these datasets in §3.2, and show that these datasets contain both correct and incorrect content word combinations which are challenging for ED algorithms. The datasets have been annotated using an annotation scheme that describes the typical confusion patterns in learner use of content words. Chapter 3 presents the annotation scheme and the results of the annotation experiment.

- In Chapter 4 we describe a simple algorithm for ED in content words that is based on the previous approaches to this task. The algorithm performs EDC by comparing the original combinations to their possible alternatives and selecting the most fluent one according to the chosen measure of collocational strength. We present the theoretical background for this algorithm in §4.1, and discuss the implementation in §4.2. We apply this algorithm to the AN and VO datasets and show that it performs poorly on learner data, and in particular on word combinations previously unseen in native corpora. The results are presented in §4.3.

- In Chapter 5 we present a novel approach to ED in content words which is based on the observation that many errors are caused by a semantic mismatch between the words within a combination. We refer to the previous work on semantic anomaly detection and show that a similar approach can help detecting errors in content word combinations in learner writing. We discuss the implementation of the models for AN combinations in §5.1 and §5.2. We present and discuss a number of measures that help distinguish between semantic representations of the correct and incorrect combinations in §5.3, and present the results in §5.4. Application of the models of compositional distributional semantics to VO combinations is discussed in §5.5 to §5.8. The approach presented in Chapter 5 shows promising results, and we discuss the directions for future work in §5.9.

- We use the output of the semantic models and the measures for detecting semantic anomaly to derive features for an ML classifier. Experiments with the ML classifier are presented in Chapter 6: we discuss the theoretical background for this algorithm in §6.1, the implementation in §6.2, and the results in §6.3. We show that an ML classifier that uses a small number of semantic features outperforms state-of-the-art approaches to ED in content words and shows good performance on this task. We discuss the performance of the algorithm in more detail in §6.4, and summarise in §6.5.

- We conclude with Chapter 7 which summarises the contributions of the current work and discusses future directions for the research on ED and EDC in content word combinations.

Chapter 2

Theoretical background

In this chapter, we give an overview of previous research on the EDC in learner writing. In §2.1, we discuss the learner corpora that are available for research including the ones that we use in this project, and review the general principles of learner data collection. The underlying principles and guidelines for learner data annotation are discussed in §2.2. We rely on these principles in annotating the data for this project.

In §2.3, we discuss the previous approaches to EDC. We consider the ML techniques used for EDC in function words and discuss whether these techniques can be reapplied to content words. §2.3.2 presents previous research on EDC in content word combinations. We note that most research in this area has either (a) assumed that errors had already been detected and performed *error correction* only, or (b) performed *writing improvement* with the goal of suggesting the most fluent word combinations to the learners rather than assessing the acceptability of the original combination, or (c) relied on manually created databases of previously seen errors and their corrections. We discuss the limitations of these approaches and conclude that none of them have addressed ED in content words.

§2.4 presents methods of distributional and compositional semantics, discusses the application of these methods to a related task of semantic anomaly detection, and outlines how these models can be applied to the current task.

Finally, we discuss how EDC systems should be evaluated in §2.5.

2.1 Learner corpora

Since nowadays learner corpora are widely available, it is possible to analyse actually occurring learner errors and perform *error analysis* (*EA*) on texts freely produced by language learners. EA has certain limitations that should be taken into account. Since it is performed on actual texts and uses a one-sided practice of "analysing out the errors and neglecting the careful description of the non-errors" (Hammarberg, 1974), it only considers occurring errors and not the "uncommitted" ones – the potentially problematic cases that learners might be unsure of even if they manage to not commit an error. Nor can this approach deal with the potential errors that are never realised due to avoidance strategies. From the pedagogical point of view, such cases that are missed through EA are of high value. It is also important to keep these issues in mind when drawing conclusions about the typical errors of language learners: for example, Leacock et al. (2014, p. 19) mention that the most frequent errors in the writing of U.S. college students (all native speakers) involve certain punctuation and sentence structure-related errors, while those error types represent only a minor portion of errors in texts written by language learners. The reason for that is not that non-native speakers master these aspects of writing more successfully than native speakers, but rather that they avoid using the structures in which such errors can occur since they are unsure of the correct use in these cases.

In spite of these issues, learner corpora allow the investigation of real 'living' learner errors as well as contexts in which they commonly occur, and EDC research benefits greatly from the availability of such corpora. The usefulness of annotated learner corpora has been acknowledged since the time of their appearance (see Dagneaux et al. (1998), Granger and Leech (1998), Tono (2003), among others). Learner corpus research is also overviewed in Granger (2007), Nesselhauf (2004) and Pravec (2002).

The availability of large annotated learner corpora in combination with the advances in ML of recent years also determined the success of statistical methods in EDC.

2.1.1 Cambridge Learner Corpus

The CLC[1] is a 52.5 million-word corpus (16 million-words in 2003) of learner English collected by Cambridge University Press (CUP) and Cambridge English Language Assessment since 1993 (Nicholls, 2003). It comprises essays produced by English language learners sitting examinations in English and written in response to examination prompts. At the moment, the CLC contains over $200,000$ exam scripts from students speaking 148 different L1s living in 217 different countries or territories. For comparison, in 2003 the corpus contained texts written by speakers of 86 L1s, and more than 15 L1s were represented with more than $350,000$ words (Nicholls, 2003). The examination scripts have been transcribed retaining all errors, and a part of the corpus (a 25.5 million-word component currently, and a 6 million-word component in 2003) has been manually error-coded by two coders using an error annotation scheme devised specifically for the CLC. The scheme contains 88 distinct error codes (Nicholls, 2003). Since a growing number of non-native speakers undertake language examinations every year, the corpus and its error-annotated section have also grown over time.

Each examination script contains meta-information about the learner, including one's L1, age, sex, education history and number of years the learner has studied English for. This information, as well as the error coding, can be used to create subcorpora representing particular subsets of learners (for example, speakers of certain L1s) or particular error types (for example, content word combinations).

The aim of error coding in the CLC was not to create a systematic error taxonomy, but rather to capture, where possible, all errors of a certain type under one heading (Nicholls, 2003). This allows further investigation of the corpus, searching for recurrent error types occurring in similar contexts. The annotation scheme is flexible enough to allow search on

[1] `http://www.cambridge.org/gb/elt/catalogue/subject/custom/item3646603/Cambridge-English-Corpus-Cambridge-Learner-Corpus/?site_locale=en_GB`

clusters of errors: for example, on all noun-related errors *en masse*. The constructions of interest can be extracted using these properties of the corpus and the annotation scheme, while more detailed annotation can be performed on the extracted data. Such a strategy has been used in this research project.

Corrections for the annotated errors are provided whenever possible. The general approach to error correction in the CLC was for the annotators to not try to 'interpret' or paraphrase the original learner's input, but to only provide a 'correct' version when there is only one clear replacement possible. The availability of the corrected version supports further functionality: for example, comparative analysis of the original (errorful) and the corrected (error-free) version can reveal certain properties of learner language, especially with respect to missing words and constructions.

The annotation and corpus-construction principles used in the CLC include:

- avoiding over-coding and 'creating' errors, when annotators are specifically instructed to not try to paraphrase, interpret or rewrite the scripts into perfect English – only the absolutely incorrect use is corrected;
- choosing codes and corrections so that they result in a minimum change in the original text and keep it as close to the original as possible. This is especially important in case of competing or ambiguous errors when several corrections and various amount of change are possible. For example, **He said me that...* could be annotated as either a wrong verb choice error (`RV`) and corrected to *told*, or as a missing preposition error (`MT`) and corrected to *said to me*. The latter, it can be argued, introduces less severe changes to the text and should be chosen in this case. Besides, it is more helpful to teach the students how to use the chosen verb correctly rather than how to avoid using a verb that they use incorrectly. This is achieved by imposing a hierarchy on the use of the error codes.

The CLC is available for research, and a subset containing the FCE scripts from the years 2000 and 2001 has been made publicly available (see §2.1.2).

Error annotation in the CLC

The majority of the error codes in the CLC are two-letter-based with the first letter representing the *general type of the error* (for example, wrong form or omission) and the second letter representing the *word class of the required word* (for example, noun). Errors are tagged using the following convention:

```
<NS type=#CODE><i>incorrect text</i><c>correction</c></NS>
```

The set of first letters in the error codes includes `F` for wrong Form used, `M` for something Missing, `R` for the word or phrase needing Replacing, `U` for word or phrase Unnecessary or redundant, and `D` for word wrongly Derived. The set of second letters includes `N` for Nouns, `J` for adJectives and `V` for Verbs, among others.

This two-letter convention makes it possible to extract contexts exemplifying certain subtypes of errors. For example, by specifying that the error code should contain `N` as the second letter, one could extract all noun-related errors, or by specifying that the error code should start with an R one could search for all words and phrases needing replacement in learner texts. A search for all the instances tagged with RN, for example, would return all the nouns used incorrectly by the language learners.

2.1.2 CLC-FCE dataset

The FCE dataset is a subset of the CLC which was released in 2011 (Yannakoudakis et al., 2011) and is publicly-available.[2] The scripts included in the dataset have been produced by learners taking the FCE exam, which assesses English at an upper-intermediate level. The scripts have been anonymised and annotated using XML, with the original metadata about the question prompts, the candidate's grades, L1 and age retained. Each script contains two essays of the length 200–400 words, written in response to tasks asking learners to write a letter, a report, an article, a composition or a short story. The scripts contain the original error annotation from the CLC (Nicholls, 2003), marks for each of the two answers and an overall score in the range 1–40 assigned to each script.

The released dataset contains 1,244 scripts from the years 2000 and 2001. The prompts are released with the dataset. A typical prompt is shown below:

> *Your teacher has asked you to write a story for the school's English language magazine. The story must begin with the following words: "Unfortunately, Pat wasn't very good at keeping secrets".*

We use this dataset for the experiments on content word ED in our project. Previously, it has been used for experiments on automatically grading (Yannakoudakis et al., 2011) and modelling coherence in ESOL learner texts (Yannakoudakis and Briscoe, 2012), as well as on grammatical EDC (Dale et al., 2012; Yuan and Felice, 2013; Felice et al., 2014), and L1 detection (Brooke and Hirst, 2011; Kochmar, 2011; Brooke and Hirst, 2012), to name just a few.

2.1.3 Other corpora

Some other learner corpora that are available for research on EDC include:

- The *NUS Corpus of Learner English* (*NUCLE*) (Dahlmeier et al., 2013). It contains about 1,400 essays from undergraduate university students totalling over 1 million words, completely annotated with error tags and corrections. The annotation and correction have been performed by professional English instructors. The tag set contains 27 error categories grouped into 13 categories, which is considered to be a sufficiently fine-grained tag set while not too complex for the annotators to efficiently apply it. Collocation errors are reported to be among the top five error categories, although the Wcip tag used for wrong word choice errors stands for *wrong collocation/idiom/preposition* and, thus, covers various types of wrong

[2] http://ilexir.co.uk/applications/clc-fce-dataset/

word choice errors, with content words as well as prepositions involved. Together, they account for 15.69% of all the errors in the corpus. Dahlmeier and Ng (2011a) reported that collocation errors alone account for about 6% of all errors in NUCLE, which makes these errors the 7th largest class of errors in the corpus after article errors, redundancies, prepositions, noun number, verb tense, and mechanics.

- *International Corpus of Learner English* (*ICLE*) (Granger, 2007) – a non-annotated learner corpus.
- *Chinese Learner English Corpus* (*CLEC*) (Gui and Yang, 2001) – a 1-million-word corpus of Chinese learner English annotated for error types but not containing error corrections.
- The annotated smaller corpora used for HOO'11 (Dale and Kilgarriff, 2011), HOO'12 (Dale et al., 2012), and grammatical EDC shared tasks (Ng et al., 2013, 2014).

2.1.4 Content word error datasets

The collection and annotation of a dataset representing learner errors in the choice of content words is an important step in this project. In spite of the availability of learner corpora in the past years, there have not been many datasets aimed at representing errors in content words. Such datasets can be extracted from learner corpora, but they do not contain any annotation specific to content word errors.

Below we list the datasets relevant for EDC in content words that have been previously released:

- Dahlmeier and Ng (2011a) presented a dataset of collocation errors extracted from real-world learner data and representing errors in one L1 (Chinese) only. They used the dataset to show that L1-induced paraphrases outperform traditional approaches to error correction based on edit distance, homophones, and WordNet synonyms. The subset of collocation errors was extracted from the NUCLE corpus using the `Wcip` error tag. After filtering, errors involving function words were removed, as well as the instances where the original or the correction was longer than 3 words. Dahlmeier and Ng (2011a) report that the collocation error dataset contains $2,747$ errors and their corrections ($2,412$ distinct pairs), and that these instances account for about 6% of all errors in the NUCLE. It was also noted that a substantial number of errors in this dataset can be traced to L1-transfer.
  Limitations:
  - The NUCLE corpus has been released, but the particular dataset of the collocation errors is not available as a separate resource and it must be extracted from the NUCLE corpus.
  - As the work is aimed at error correction rather than detection *and* correction, it only represents errors and does not contain originally correct examples. It is suitable for exploring the correction step, but real learner data contains both correct and incorrect instances, and is typically skewed towards correct examples.

  - The dataset exemplifies only one L1.

- Vecchi et al. (2011) focused on detecting semantic anomaly in AN combinations and compiled a dataset of ANs not encountered in native corpora of English, including the Web-derived ukWaC corpus,[3] a mid-2009 dump of the English Wikipedia[4] and the *British National Corpus* (*BNC*).[5] First, they focused on 30 randomly chosen adjectives, combined them with $8K$ most frequent nouns, and then randomly selected 100 ANs for each of the adjectives ($3K$ ANs in total). Two authors went through this list, marking ANs as semantically acceptable, intermediate or semantically anomalous. As a result, they collected a set of 456 ANs where both judges agreed the AN was odd and 334 where both judges agreed the AN was acceptable. Then 5 adjectives were discarded for either technical reasons or for having less than 5 agreed deviant or acceptable ANs. The final set contained 413 deviant AN combinations (e.g., *academic bladder*, *blind pronunciation*, *parliamentary potato* and *sharp glue*), and 280 acceptable but unattested AN combinations (e.g., *vulnerable gunman*, *huge joystick*, *academic crusade* and *blind cook*).[6]
  We note that this dataset exemplifies a closely related problem in native English – semantic anomaly. The errors in content word combinations committed by language learners often stem from a semantic mismatch between the chosen words, a phenomenon analogous to 'semantic anomaly' in the language of native speakers. However, we also note that native speakers know how to choose content words in their language and are usually assumed to not commit errors, while language learners might only unintentionally produce a semantically anomalous combination, as they still try to transmit a sensible message. Therefore, we consider that this dataset cannot be used for research in content word ED, since it exemplifies a similar but not exactly equivalent problem.

As there has been no previously released dataset of content word combinations annotated with respect to learner errors, we consider this to be an important contribution of this work, filling a gap in learner language research.

2.2 Learner data annotation

According to Corder (1974) and Ellis (1994), EA comprises four steps: *error data collection*, *error identification* – selection of the text spanning an error, *error classification and description*, and finally *explanation and correction*.[7] Error annotation follows a similar procedure: errors are readily available in learner text, and annotators are asked to *identify*, *classify* and *correct* them.

Certain annotation principles can be adopted from previously employed annotation schemes. For example, the annotation scheme used for the CLC was primarily aimed at grouping

[3] http://wacky.sslmit.unibo.it

[4] http://en.wikipedia.org

[5] http://www.natcorp.ox.ac.uk

[6] The sets can be downloaded from http://www.vecchi.com/eva/resources.html

[7] James (1998) suggests a similar approach (see §1.2.1).

errors of a certain type under one heading (Nicholls, 2003). The annotators were also instructed not to paraphrase the original content, and to use a tag hierarchy to resolve ambiguous cases. Annotators of the CLC (Nicholls, 2003), as well as annotators of the NUCLE (Dahlmeier et al., 2013), were instructed to identify a minimal text span for an error, in order to minimise changes made to the original text to correct it. In what follows, we discuss some other issues relevant for data annotation.

Lüdelig et al. (2005) advocate a multi-level standoff rather than a flat token-tag architecture that is currently used in most learner corpora. They demonstrate that standoff annotation allows for encoding of several competing hypothesis at a time, as well as tagging of interacting and overlapping errors, which is useful when a number of different errors are present in a learner corpus. For example, both *I like reading* and *I like to read* are suitable corrections for **I like read*, even though only an annotation format allowing for multiple hypotheses can encode both corrections at once. The recent shared tasks on grammatical EDC have also used standoff annotation (Ng et al., 2013, 2014). Encoding several competing hypotheses helps cover multiple possible corrections, but we note that for detection the number of correction hypotheses is not directly relevant. We focus on a particular error type and the issue of overlapping errors does not arise in our dataset either.

The next question is how many annotators should be used for comprehensive data annotation. Dahlmeier et al. (2013) note that error annotation is not an easy task even for trained annotators, which is confirmed by the low inter-annotator agreement reported. Madnani et al. (2011) note that most usage errors such as those in articles and prepositions are a matter of degree rather than simple rule violation as, for example, in the case of number agreement. As a result, two native speakers might have different judgements of usage, and this makes error annotation a difficult task and leads to low inter-annotator agreement. This is the main argument for using multiple annotators for gold standard error annotation in learner data, because multiple judgements can be aggregated in a single score representing the degree of correctness.

Usually, learner corpora are annotated by one or two professional annotators, since annotation is both time-consuming and expensive. Crowdsourcing is suggested as a viable alternative by Madnani et al. (2011). They focused on extraneous preposition detection, and used a corpus of $1,000$ sentences with a 50% error rate which was first annotated by native speakers, and then by untrained annotators – Turkers – using a crowdsourcing service. Madnani et al. (2011) found that, on average, only 3 untrained raters were enough to match the experts: when a majority vote was used for just 3 untrained annotators, their agreement with any one of the 3 expert raters was, on average, 0.87 with a *kappa* (κ) of 0.76 which is on a par with the inter-expert agreement and κ. In addition, crowdsourced annotation was both cheaper and faster.

Tetreault et al. (2013) explored how reliable crowdsourced annotations are, and concluded that different tasks require different number of Turkers: 3 Turkers are needed to match the judgements of 2 expert annotators on fill-in-the-blank task for prepositions, while 13 Turkers are needed to match 3 experts on the preposition ED task. For collocation ED, 4 Turkers are needed to match 4 experts; however, when Turkers are first tested using some quality control annotation questions, the number of Turkers required drops to 3. Higher levels of agreement were obtained on higher frequency n-grams than on lower frequency

n-grams, and on "cleaner" error-free sentences than on "noisy" sentences, though experts proved to be more tolerant to noise than Turkers.

Madnani et al. (2011) also suggested evaluating ED in a *graded* manner and modifying standard precision and recall measures to incorporate distribution of correctness obtained via crowdsourcing in order to make the measures fairer and more stable indicators of system performance. The main motivation is that the standard measures and binary error classification can only evaluate instances as either correct or incorrect, making the ED too coarse-grained for what is a complex phenomenon. This way, all errors are treated as equally "bad" and all correct uses are treated as equally "good", so valuable information about the acceptability of usage is discarded. Madnani et al. (2011) argue that it is fairer to represent correctness as a scale rather than a binary classification, and that cases that are most controversial for human annotators should contribute to the evaluation measures differently from those on which annotators agree. A system's output cannot be considered entirely right or entirely wrong if humans cannot consistently decide whether a case is an error. The standard precision and recall measures, as Madnani et al. (2011) note, can over- or under-estimate the real system's performance. They propose using weighted measures instead, which are also more stable, as the majority vote used for the binary case can very easily change the polarity of judgement from error to correct, or vice versa. The proportion of disagreement between multiple annotators is a valuable piece of information since it highlights the difficult cases on which the system cannot be expected to perform as well as on the cases of clear errors.

Lee et al. (2009) also found that annotators often identify more than one possible correction in an experiment on the use of articles and noun number, and according to this experiment, an EDC system's performance may be underestimated by 18% or more if multiple possible corrections are not taken into account. They also noted that the proportion of agreement among the annotators should be used for measuring the system's performance: the system should be expected to perform well on the nouns with strong agreement.

Reliability of human judgements has also been tested in an experiment on preposition error annotation by Tetreault and Chodorow (2008a). They focused on preposition use as they show that humans tend to find this task difficult, possibly, due to the variety of linguistic functions prepositions serve. They concluded that rating preposition usage in either native or non-native texts is a task that has surprisingly low inter-annotator reliability and thus greatly impacts system evaluation. When the annotators were asked to fill in the blank in a sentence with the best-fitting preposition, the inter-annotator agreement between the two annotators was only about 76%, and ranged from 74% to 78% when each rater's selection was compared to the original preposition removed from the text. When the prepositions suggested by each rater were examined, it was confirmed that they were also licensed by the context and were acceptable alternatives to the original prepositions. Additionally, they also trained a maximum entropy classifier that suggested prepositions for another 200 sentences from Encarta, and presented these sentences to the human annotators with a choice of two prepositions in random order: one was the original preposition, and another one was the system's suggestion. The results showed that both raters considered the preposition suggested by the system equally good or better than the original one 28% of the time. This means, that automatic evaluation based on comparison to a single preposition in the gold standard in contexts licensing multiple alternatives can underestimate a system's performance by as much as 28%, and these cases are not system errors. Finally, they also found that there is a 10% difference in precision and a 5% difference in recall between the two system/rater comparisons. The authors conclude that there is a potential to over- or under-estimate precision by as much as 10% using only one annotator.

We maintain, however, that the need for multiple corrections does not necessarily show the need for multiple annotators, as multiple corrections can be suggested by a single annotator. For ED, the number of suggested corrections is also not directly relevant. In spite of the benefits of using multiple annotations, it is still not clear how to use this feedback and whether it is what learners expect from an EDC system. Learners are used to binary evaluation – correct usage versus error – and expect to see errors flagged in the text: being informed that something is an error, they know what should be changed in the text. It is not clear whether they would actually benefit from seeing that something is partly acceptable (for instance, knowing that a particular adjective used with a particular noun might or might not be an error depending on the annotator), or whether they would rather prefer knowing that the word is inappropriately chosen and should be corrected. Evaluation on graded rather than binary annotation has so far been intrinsic (system-oriented), while to show the usefulness of such an approach one should also conduct an extrinsic (user-oriented) evaluation.

2.2.1 Content word error annotation

It is useful to have both a representative dataset and a comprehensive annotation scheme for each specific task. To the best of our knowledge, there is only one error annotation scheme previously devised for content word combinations and presented in the literature.

Ramos et al. (2010) devised an annotation scheme for content word combinations, or collocations, which they defined following the common lexicographic tradition (Mel'čuk, 1998): they assumed that a collocation is a restricted binary co-occurrence of lexical units between which a syntactic relation holds, and that one of the lexical units (the *base*) is chosen according to its meaning as an isolated unit, while the other (the *collocate*) is chosen depending on the base and the intended meaning of the co-occurrence as a whole, rather than on its meaning as an isolated lexical unit. This definition is the closest to the definition of the content word combinations addressed in this thesis. Ramos et al. (2010) error-annotated collocations extracted from the CEDEL2[8] – a 400,000-word corpus of essays written in Spanish by native speakers of English. A detailed collocation error typology and an error annotation scheme were devised to annotate the errors in this data. The main motivation was the need for a detailed collocation error classification for facilitating development of both EDC systems and targeted learning exercises. The annotation scheme is very fine-grained and encodes the errors along three dimensions.

The first dimension encodes whether the error concerns the collocation as a whole, or only one of the constituent words (the base or the collocate), and comprises three error tags. The second dimension models the analytical (linguistic) error analysis and characterises

[8] `http://www.uam.es/proyectosinv/woslac/cedel2.htm`

errors as errors in register, lexis or grammar. This level covers a wide variety of phenomena, including incorrect lexical choice or errors in the chosen grammatical category of number or gender on one of the words within a collocation. This level comprises 14 distinct error tags. The third dimension models the explanatory analysis, and seeks to define possible error sources. The 11 error tags at this level are grouped into *interlingual* (L1 transfer) and *intralingual* (L2) errors for register, lexis and grammar. The preliminary results reported have shown that most errors are those of lexical choice, with the collocate rather than base being chosen incorrectly.

This annotation scheme provides some valuable information about the subtypes of the collocation errors and possible reasons for such errors. However, we note that, first of all, it might be too fine-grained for practical use. The obvious danger in employing a too fine-grained scheme is information overload for the annotators, when, as a result, they achieve lower agreement and the annotation becomes less reliable. The annotation scheme that we have devised is aimed at being comprehensive, while also being clear and manageable. Secondly, we note that certain error types, such as word order violation within the collocation, or agreement in gender and number between an adjective and a noun, occur in Romance languages like Spanish but not in non-Romance language like English. Therefore, this annotation scheme is not language-independent, while we aim at a general scheme that can easily be extended and applied to other languages as well as various types of content word combinations. Finally, the explanatory level of annotation seems to be the most challenging for the annotators who are asked to distinguish between inter- and intra-lingual errors – a task that becomes harder when multiple L1s are involved. In Ramos et al. (2010), the data was produced by native speakers of a single L1 which made it possible to detect L1 transfer, while our data covers 86 L1s. Therefore, we make no effort at detecting L1 transfer in a systematic way.

2.2.2 Inter-annotator agreement

The purpose of measuring inter-annotator agreement on the learner datasets is two-fold. First, it helps assess the quality of the annotation scheme and the guidelines: if the scheme has sufficient descriptive power and is not ambiguous, and the guidelines are clear and not contradictory, then the annotators might be expected to have high agreement when using the annotation scheme with the provided guidelines. Second, inter-annotator agreement is a measure of the difficulty of the task itself: it is usual to set inter-annotator agreement as an upper bound for the algorithm, as the automated system might not perform well on a task that is naturally hard for humans.

Inter-annotator agreement is usually measured using Cohen's *kappa* (κ) coefficient (Cohen, 1960). The procedure of calculating inter-annotator agreement is aimed at determining how reliable the judgements are, or determining "the degree, significance, and sampling stability of their agreement" (Cohen, 1960). It is based on three independence assumptions: the units for annotation are independent, the categories to be assigned are independent, as well as mutually exclusive and exhaustive, and the annotators operate independently while being considered equally competent and unbiased in their judgements. The coefficient proposed by Cohen takes into account not only the observed agreement between the annotators p_o, but also the agreement that is expected by chance p_c estimated by finding the joint probabilities of the marginals. The coefficient κ is "the proportion of chance-expected disagreements which do not occur, or alternatively, it is the proportion of agreement *after* chance agreement is removed from consideration" (Cohen, 1960):

$$\kappa = \frac{p_o - p_c}{1 - p_c} \tag{2.1}$$

κ expresses the importance of the items agreed upon between annotators $(p_o - p_c)$, normalised by the disagreement between annotators that would be expected by chance $(1 - p_c)$:

$$\kappa = (\#\ \textit{items agreed upon above chance})/(\textit{expected disagreement})$$

If the two annotators' judgements are distributed among two categories as shown in Table 2.1, then

$$p_o = \frac{a + d}{N} \tag{2.2}$$

and

$$p_c = (\frac{a + c}{N} \cdot \frac{a + b}{N}) + (\frac{b + d}{N} \cdot \frac{c + d}{N}) \tag{2.3}$$

Annotators		Annotator$_1$		
	Categories	Cat$_1$	Cat$_2$	Total
Annotator$_2$	Cat$_1$	**a**	b	$a + b$
	Cat$_2$	c	**d**	$c + d$
	Total	$a + c$	$b + d$	N

Table 2.1: Judgements by two annotators.

The interpretation of κ values is given in Table 2.2 following Landis and Koch (1977).

Kappa (κ)	Agreement
< 0.00	less than chance agreement
0.00	chance agreement
0.01 – 0.20	slight agreement
0.21 – 0.40	fair agreement
0.41 – 0.60	moderate agreement
0.61 – 0.80	substantial agreement
0.81 – 0.99	almost perfect agreement
1.00	perfect agreement

Table 2.2: Interpretation of κ values.

Finally, to show that the obtained κ values are statistically significant, one can estimate the standard error of κ and test the null hypothesis that κ arose in sampling from a population of units for which $\kappa_p = 0$ (Cohen, 1960):

$$\sigma_{\kappa_0} = \sqrt{\frac{p_c}{N(1-p_c)}} \tag{2.4}$$

and

$$z = \frac{\kappa}{\sigma_{\kappa_0}} \tag{2.5}$$

Cohen's κ is only applicable to measuring agreement between two annotators. For multiple raters, there are other measures, for example Fleiss' κ (Fleiss, 1971). However, since this measure has not been widely used in previous studies on learner data annotation and since there is no widely accepted scheme for interpreting the values of Fleiss' κ, it is harder to make meaningful comparison. Instead, we report the Cohen's κ values for each pair of annotators, as well as the average κ value.

The reported inter-annotator agreement for EDC is usually not very high, which shows the natural complexity of the task. For example, the NUCLE was annotated using a taxonomy of 27 error tags in 13 categories, and the set of 96 double-annotated student essays was used to estimate inter-annotator agreement (Dahlmeier et al., 2013). The reported Cohen's κ values are quite low: 0.3877 for ED, 0.5484 for error classification given the annotators agreed on the error identified, and 0.4838 for exact agreement on both error classification and correction given the annotators agreed on the error identified. According to κ value interpretation, the κ score for ED can only be considered fair, while that for classification and exact agreement are moderate. Dahlmeier et al. (2013) note that the annotators found it harder to agree on whether a word is grammatically correct or not than to agree on the type of the error and the correction. They conclude that grammatical ED is a difficult task even for trained annotators.

2.3 Error detection and correction

Leacock et al. (2014) and Tetreault et al. (2013) list the following approaches to EDC:

1. *Rule-Based Approaches* were popular in the early days of EDC systems development due to the lack of large corpora to train the systems and ease of using rules for certain error types. Even today some error types, such as subject-verb agreement, are easier to handle with manually-crafted rules. However, due to the complex nature of usage errors, hand-crafted rules cannot be applied to errors in articles, prepositions and content words.

2. *ML Classifier Approaches* have become popular due to the availability of large training resources and advances in ML of recent years. Earlier systems were trained on native English corpora – on millions of examples of correct usage – with various features explored, including syntactic, lexical, n-gram and even semantic features (Felice and Pulman, 2008; Gamon et al., 2008; Tetreault and Chodorow, 2008b). Later approaches have both explored more sophisticated ML approaches (Dahlmeier and Ng, 2011b), and used learner data as well as artificially generated errors (Rozovskaya

and Roth, 2010a,b; Yuan and Felice, 2013; Felice and Yuan, 2014; Felice et al., 2014). These methods can handle certain types of errors – for example, those in closed class words with finite confusion sets – but are less suitable for content word combinations.

3. *Language Modelling (LM) Approaches* are also based on large data sources used for learning language models. Within this framework, the target word is substituted with its alternatives and the LM scores are derived for the original text as well as for the alternatives. The word yielding the highest probability is chosen as the correct word in the given context. A number of systems used language models to rank the suggestions of the ML classifiers (Gamon et al., 2008; Chodorow et al., 2010; Felice et al., 2014), or in combination with an ML classifier (Gamon, 2010). Many approaches in the recent shared tasks on grammatical EDC have been based on or used LMs (Ng et al., 2013, 2014), but none of them addressed content words. This approach is problematic because it is subject to selecting and comparing alternatives, and very inefficient for content words for which the number of possible alternatives can be high. In addition, the collection of large amounts of high quality data for training LMs to address EDC in content words is an issue.

4. *Web-based Approaches* make use of large scale Web data. They try to mimic the approach often used by language learners who search for their original constructions on the Web and compare Web counts with those for the alternatives using the counts as a proxy of correctness. Some methods based on this idea have addressed content word combinations (Hermet et al., 2008; Yi et al., 2008). In *Microsoft Research ESL Assistant*, Web search was used to provide the learners with examples for the original and the alternatives (Chodorow et al., 2010). This approach suffers from the fact that Web counts can be very unstable from one day to another, and the counts around the target word can be sparse reducing the reliability of the approach (Kilgarriff, 2007).

5. *Statistical Machine Translation (SMT) Approaches* cast the EDC task as that of transforming a "noisy" English sentence into a fluent English sentence. Earlier work used SMT for small sets representing some error types (for example, mass noun errors (Brockett et al., 2006)), while later SMT has been used to address a wide variety of errors. For example, a number of the teams participating in the recent shared tasks on grammatical EDC used SMT-based systems (Ng et al., 2013, 2014), but none of them reported good results on the content word errors in particular, and a possible reason for that is that SMT approaches are good for local sequences of errors, but use of local context is not effective for sparse content word error types.

2.3.1 Function words

Most often, EDC for function words is concerned with articles and prepositions. These words are, on the one hand, frequently used in English, and, on the other, are some of the most difficult elements of English for language learners (Chodorow et al., 2010; Leacock et al., 2010, 2014). The high number of errors in function words is related to both their frequency in English itself and to the difficulty in mastering these elements of language.

The latter can be explained by the complexity and variability of article and preposition usage in English: the correct choice of an article can depend on the discourse structure (first or subsequent mention in text), countability/non-countability, as well as the noun and the words modifying that noun. For example, **a damage* is clearly unacceptable, while *a little damage* is fine (Chodorow et al., 2010). The use of prepositions is determined both by the governing and the dependent word (*I will come by*, *I came from London*). It is the combination of various factors that governs the choice of the correct article and preposition, and Tetreault and Chodorow (2008a) have shown that the judgements of two trained native speakers can differ by as much as 10% when marking preposition usage in non-native writing as correct or incorrect, and as much as 24% on the task of filling in the best preposition. This shows that the function word system of English is complex and even native speakers show substantial disagreement on that. For comparison, some other grammatical categories like subject-verb agreement are less controversial.

Non-native speakers experience additional problems in learning English articles and prepositions caused by the mismatch of these systems in English and their L1s. Native speakers of languages which do not have articles are even more confused about their use in English since they lack a background understanding of the article system.

Since errors in function words are some of the most common error types in learner writing (Dalgish, 1985; Leacock et al., 2010), it is important for any EDC system to be able to deal with them. Certain properties of these errors facilitate the use of ML approaches for their detection and correction and help finding the relevant features.

As function words belong to closed classes, the set of possible corrections is limited by the size of the function word set. Since errors in function words are systematic and highly recurrent, in practice, each article or preposition has an even smaller number of appropriate alternatives. This can be illustrated with the following examples on (1) article and (2) preposition errors:

(1) I am *0*/a* student.

(2) Last October, I came *in*/to* Tokyo.

In (1) an EDC system would consider {*a, an, the*} as possible corrections for the missing article. To correct the preposition *in* in (2), an EDC system would consider {*on, from, for, of, about, to, at, with, by*}. According to Leacock et al. (2010), the usage of the 10 most frequent prepositions accounts for 82% of all preposition errors in the CLC, and according to Chodorow et al. (2010), the top 10 prepositions in English account for roughly 91% of preposition usage, so the set of possible confusions for prepositions is often limited to the 10 most frequent ones. Among these prepositions, *at* or *to* would have a higher chance to be appropriate corrections than others as these are more frequently confused with *in*. Confusion sets can be learnt from learner texts, and the probabilities can be set up according to the distribution of the confusing words (Rozovskaya and Roth, 2011).

The usual approach to EDC in function words is to cast it as a multi-class classification task, with the number of classes equal to the number of target corrections – for example, 4 for articles and 10 for prepositions. Detection and correction can occur simultaneously:

an error is detected when an EDC system suggests using a word different from the one originally used by the learner, and the suggested word can be used as a correction. For example, the context '*We sat __ the sunshine*' might be assigned to the preposition class '*in*' but not to the class '*at*'. If this context occurs with '*at*' filling the blank, the classifier marks it as an error (Chodorow et al., 2010).

Context is highly informative for EDC in function words: since these errors are recurrent and the number of possible confusion patterns is finite, there is much to be learned from the surrounding context. The general approach is to represent each occurrence as a feature vector: for example, a context of *I am* and *student* or a similar noun requires the use of an indefinite article, while the only correct preposition to relate a verb of movement like *come* to a locative like *Tokyo* is *to*. Therefore, the preceding and following words within some context window, their PoS tags, the words grammatically related to the function word and other types of information extracted from the surrounding context can be used as features. In general, the number of possible feature values can grow quickly: for example, the feature *following word* for an article use can include all the words like *student*, *teacher*, *driver*, *banker* and the like for the indefinite article. In practice, it is possible to group those into one semantic class of nouns since it is not the particular noun that defines the use of the indefinite article in this case, but rather the class of nouns denoting *occupations*.

2.3.2 Content words

Errors in content words, unlike those in function words, are less systematic, with more diverse reasons for the confusion. The contexts are sparser, and as a result, less useful for feature extraction. The number of classes or corrections cannot be reduced to a finite set as in the case of the function words. As a result, it is much harder to cast EDC for content words as a classification problem with a fixed number of classes and to apply ML classifiers.

In spite of the fact that errors in the choice of content words have not received much attention in previous years, their importance has been widely acknowledged. Content words are responsible for content transfer, so violation of conventions leads to misunderstanding: for example, if a learner uses **big conversation*, do they mean an *important conversation* or a *long conversation*? The original meaning in the incorrect combination is distorted.

We begin with an overview of the typical errors in content word combinations. The scheme devised for content word annotation in this project is described in Chapter 3, but we note that we based the scheme on our observations of learner data, as well as on the error categories overviewed in this section. We then discuss the previous approaches to EDC for the related error types.

Errors in content word combinations

James (1998, p. 142) refers to content word choice errors as lexical errors, and notes that learners believe that vocabulary is very important in language learning to the point that language learning itself would sometimes be equated with knowing language vocabulary.

Following some previous work in this area, James (1998, p. 144) distinguishes between *formal* and *semantic* lexical errors. The cases described under these categories are useful for the error taxonomy that we build in this research.

Formal errors are cases of confusion where learners mix up words having similar stress patterns, number of syllables, phonemes in common and so on. It is hard to distinguish between the cases when learners are simply confused by superficial word similarity but are aware of the difference between the confusables, and those which result from ignorance. When choosing a content word, learners generally follow one of two strategies: either they think that they know the word and they use it, or they are not sure they know the word and they employ a lexical *communication strategy* – they avoid the concept or try to paraphrase. *Formal* errors describe the cases when a word used is a real existent word of language, and the substitute resembles the target word in form but not necessarily in meaning (for example, *classic* and *classical*), though it might do so accidentally (for example, *economic* and *economical*).

Formal errors can result from a resemblance between words in the L2 as well as in the L1. For example, a confusion in *He wanted to *cancel|conceal his guilt* results from resemblance of the two words in the L2, while *Can I *become|get* (from German 'bekommen') *a beefsteak?* results from that in the L1. The German verb *bekommen* in that case is a false friend of the English verb *become*. Other L1-effects include literal translation from the L1 to English and calque as in **sleep suit* ('pyjamas' from German *Schlafanzug*), or *America has *made profit* ('benefited' from German *Profit machen*). L1-related confusions are of high relevance for our research, but in practice it is hard to detect all such cases given the variety of L1s in our data.

Under *semantic* errors, James (1998, p. 151) lists *confusion of sense relations* and *collocational errors*. Confusion of sense relations occurs because, as multiple neurolinguistic studies demonstrate, humans store words in the mental lexicon in terms of sense relations between them. The main types of confusion caused by such relations can be summarised as:

- using a more general term where a more specific one is needed (hypernym for hyponym). This case has been mentioned in §1.2. The result is underspecification of meaning as in:
  - *The flowers had a special *smell|scent/perfume.*
  - *The village women *washed|scrubbed the steps.*
  - *Capitalism made America *big|great/powerful.*
- using too specific a term (hyponym for hypernym):
  - *The *colonels|officers live in the castle.*
- using the less apt of two co-hyponyms:
  - *a decision to *exterminate|eradicate dialects*
  - *She is my *nephew|niece.*

- using the wrong one from a set of near-synonyms:
  - *a *regretful|penitent/contrite criminal or sinner*

Learners store these related words in their lexicon but fail to distinguish between them.

Collocational errors include a substitution of the word within a phrase that violates idiomaticity of that phrase. James (1998, p. 152) distinguishes between three degrees of collocation:

- semantically determined word selection: one can say *crooked stick* but not **crooked year* because in the world as we know it years cannot literally 'be crooked';
- combinations with statistically weighted preferences: one can say 'an army has suffered *big losses*' but *heavy losses* is preferred;
- arbitrary combinations: we *make an attempt* and *have a try* but not **make a try* and **have an attempt*, even though *attempt* and *try* are very close semantically.

In addition to the causes reviewed above, some errors in VO combinations can be related to the use of de-lexicalised verbs such as *take* as in *take your time*, *make*, *keep* and other 'light' verbs. This gives rise to a significant number of errors (Chang et al., 2008).

Another potential source of confusion, which is often aggravated by L1-transfer, is the fact that certain actions can be literally described with the words that are chosen incorrectly in English. The confusion might originate in one's L1, where the action is described using different terms, but the fact that the words literally match the action could further mislead the learner: for example, *doing homework* often implies *writing*, so the learner might not suspect that using a verb *write* with the noun *homework* would produce a miscollocation. An L1-transfer from Chinese results in a miscollocation **eat a medicine/pill*, yet the act of taking a medicine could be said to resemble that of eating.

The learner's L1 is, in general, a rich source of confusion for language learners. Errors can be caused by overgeneralisation and assumption of one-to-one correspondence between lexical items in two languages. For example, German is a language close to English and German *hoch* corresponds to English *high*. Yet, while *hoch* collocates equally with the nouns *Risiko* ('risk') and *Alter* ('age') in German, only the combination *high risk* is idiomatic in English while *?high age* sounds strange.

Some errors may have multiple causes: for example, the error in *My watch does not *walk well* can be attributed to the phonological confusion between *work* and *walk*. However, knowing that the learner's L1 is French and that in French the analogous phrase is *Ma montre ne marche* (*'walk'*) *pas bien*, one could assume that the confusion is lexical as well. In such cases, an error annotation scheme can have several error tags assigned, or a hierarchy among the error tags could be introduced to resolve ambiguous cases.

Adherence to the collocational conventions of the L2 contributes greatly to one's idiomaticity and fluency, and some researchers claim that not doing so announces one's 'foreignness': for example, Nation (2001, p. 321) notes that "language knowledge is collocational knowledge", and Kjellmer (1991) and Aston (1995) note that the use of collocations or prefabricated lexical units facilitates communication and comprehension, and makes speech more native-like. Some researchers also note the strong correlation between collocation knowledge and proficiency level (Shei and Pain, 2000).

Yet, we contend that ED in content words should primarily be concerned with accuracy rather than fluency: we assume that in cases when the original word combination chosen by the learner is acceptable in English, an EDC algorithm should not correct it to an idiom or a more fluent collocation simply because this collocation is more frequently used by native speakers. We believe that if the learner is encouraged to simply memorise idioms and collocations rather than try to understand how words combine there is little to be learned about the language itself.

There is also no perfect consensus on the definition of a collocation: to name just a few definitions, collocations are defined as a sequence of words or terms which co-occur more often than would be expected by chance, as "words co-occurring within a short space of each other" (Sinclair, 1991, p. 170), as "arbitrary and recurrent word combinations" (Benson, 1990), and as strings "of specific lexical items that co-occur with mutual expectancy" (Nattinger and DeCarrico, 2001, p. 36).

We define the task as ED in *content word combinations* rather than *collocations*, highlighting the fact that we do not restrict ourselves to proper collocations only. Most of the previous work on EDC in content words has referred to the errors as 'collocation errors', and primarily focused on such cases as the use of **powerful tea* and **strong computer* instead of *strong tea* and *powerful computer* (Leacock et al., 2010, 2014). The algorithms aiming to detect failed attempts at using collocations rely on the idea that *powerful* and *strong* are semantically similar and might be confused by learners, and that **strong computer* has a much lower collocational strength than *powerful computer*. We note that this approach could only address a subset of learner errors in the use of content words, and that in real learner data neither original combinations nor their corrections might be proper collocations.

Previous approaches to EDC in content words

Errors in the choice of content words are the third most frequent error type (Leacock et al., 2010), and are the most prevalent error type marked by teachers using the *Intelligent Web-based Interactive Language Learning* (*IWiLL*) platform (Wible et al., 2003). According to Leacock et al. (2010, p. 25), not all types of content word combinations are equally challenging for language learners, and the most problematic are combinations with verbs. Several studies confirm this observation: Nesselhauf (2003) for German students, Wible et al. (2003) and Liu (2002) for Taiwanese and Chinese students, Yi et al. (2008) for Japanese, Korean and Chinese students. Liu (2002) reports that 87% of lexical miscollocations in the English Taiwan Learner Corpus (English TLC)[9] are verb–noun combinations, with 96% of those being due to misuse of verbs. 56% of these miscollocations can be corrected with a semantically related word – a synonym, hypernym or troponym – using *WordNet*.

[9] http://lrn.ncu.edu.tw/Teacher%20Web/David%20Wible/The%20English%20TLC.htm

Most approaches to collocation EDC have used the following three-step algorithm:

1. *Step 1*: Identify the candidate miscollocations in learner writing;
2. *Step 2*: Find the appropriate alternatives (for example, synonyms) for the words within the combination and generate alternative word combinations;
3. *Step 3*: Compare alternative word combinations to the original with respect to their fluency.

Most previously used methods rely quite heavily on databases of collocations and miscollocations, as well as resources for finding alternatives. They are also dependent on the measures detecting the fluency of combinations. We note that fluency does not necessarily coincide with correctness – some acceptable word combinations may be less fluent than others – therefore, the systems are prone to overcorrection. Finally, the two steps of detection and correction are merged, and detection is dependent on the quality of the alternative suggestions: if a more fluent alternative is found it is suggested by the system even if the original combination is correct but less fluent, while an originally incorrect combination might not be detected if either it is not identified at *Step 1*, or there is no better alternative found by the system in subsequent steps.

Shei and Pain (2000) addressed the miscollocations in VO combinations in Chinese writing. A set of correct examples was collected from the BNC using z-scores, and acceptable English collocations were ordered by their collocational strength. Therefore, idioms and collocations were ordered before free word combinations, as they have higher z-scores. The goal was to suggest the most fluent combinations. A database of miscollocations was collected from the learner corpus of English texts written by post-intermediate Chinese learners of English. The collocations were detected by selecting the VO combinations with the given verbs and checking them against the reference database of correct collocations. The alternatives for the candidate miscollocations were generated using the synonyms extracted from WordNet. The most fluent combination was then suggested as the correction to the learner. If the combination was not found in either the corpus of correct collocations or the corpus of miscollocations, and the learner accepted one of the suggested corrections consisting of the synonyms, the original combination was automatically added to the corpus of miscollocations allowing the system to self-propagate. This approach was used as part of a self-tutoring system, but the results were not reported.

In Wible et al. (2003), the miscollocations, including VO and AN combinations, were annotated by teachers in the data collected through the *IWiLL* platform and from the *English Taiwan Learner Corpus* (*EnglishTLC*), which at the time contained about $260,000$ words in the annotated part. The set of miscollocation–correction pairs identified and provided by the human annotators was stored for further reference. Clearly, these pairs can be used to correct errors with high precision but low recall. To improve the recall of the system, a bootstrapping method was used. It was shown that the method can automatically detect 66 candidate miscollocations in EnglishTLC, 63 of which were indeed miscollocations, which resulted in precision of 95.5%, but recall was not reported.

Chang et al. (2008) addressed miscollocations in VO combinations produced by Taiwanese learners of English and focused on those which can be traced back to L1 interference only.

Strong L1 effects on the errors committed by Taiwanese learners were first noted by Liu (2002) who reported that in the set of 233 VO miscollocations at least 84% can be explained by L1 interference. The method relied on three types of resources: the error database consisted of 233 VO miscollocations collected by Liu (2002), the alternatives were searched for through translational equivalents in a bilingual Chinese-English dictionary, while the acceptable English collocations were collected from the BNC. The system first focused on the VO miscollocations, and once they were found in text, a number of alternatives was created using translational equivalents, and assessed with respect to their frequency and collocational strength using the BNC. The alternatives were then presented to the learners ordered by their frequency and collocational strength. If the learner accepted one of the corrections, the error–correction pair was added to the error database for future use, allowing the system to self-propagate. Chang et al. (2008) reported that when their system was applied to the set of miscollocations collected by Liu (2002) and 200 manually checked correct VO collocations from the *Sinorma* corpus, the system performed ED with an accuracy of 93.9%. In the correction step, the top 10 corrections suggested by the system were compared to corrections provided by teachers, and it was shown that the system performed with 84.0% precision and the *mean reciprocal rank* (*MRR*)[10] of 0.50. Manual qualitative analysis of the system's corrections showed that, in fact, 94.1% of the system's suggestions were acceptable, which increased the MRR to 0.66.

Futagi et al. (2008) focused on automatic EDC of collocations of different types including AN and VO combinations, as well as longer collocations. The $1,446$ collocations were detected in a set of 300 randomly selected essays written by non-native English examinees for TOEFL using PoS information and a pattern-matching algorithm. The method compared the original combinations with the alternatives, which were generated using synonyms extracted from WordNet and Roget's Thesaurus, as well as morphological variations of the words within the combinations. To suggest the most fluent alternative, a reference database collected from the Lexile corpus[11] and SourceFinder corpus[12] was used and the candidates were assessed using rank-ratio statistics. The performance of the automatic tool was compared to the judgements of two human annotators, and it was demonstrated that the tool showed higher performance on the correct collocations (*F-measure* = 0.83-0.84) than on the incorrect ones (*F-measure* = 0.31-0.34). Futagi et al. (2008) also concluded that the performance approaches the upper bound on this task, as the *F-measure* estimated on the examples judged to be correct by both annotators was 0.91, while on those judged to be incorrect it was only 0.34 illustrating the natural difficulty of recognising incorrect content word combinations.

Park et al. (2008) proposed an assistive tool *AwkChecker* for EDC in collocation errors in non-native writing. AwkChecker was implemented within the context of a web-based text editor that flagged collocation errors and suggested alternatives from which a user could select the most appropriate one. The approach to EDC was based on language modelling. AwkChecker included a training interface that analysed the underlying corpus, which can be a general corpus of English or any domain-specific one, and built a set of n-grams

[10] The MRR is assessed as the arithmetic mean of the inverse ranks for the first answers returned by the system that coincide with the corrections provided by a human annotator.

[11] `http://www.gadoe.org/Curriculum-Instruction-and-Assessment/Assessment/Pages/Lexile-Framework.aspx`

[12] `https://www.ets.org/Media/Research/pdf/RR-02-12-Passonneau.pdf`

with the assigned probabilities. Next, AwkChecker analysed the input text and for every phrase in question identified whether the phrase was a collocation error by comparing it with the collected n-grams. Meanwhile, alternatives for the content word combinations were created using synonyms from WordNet. If the phrase was flagged as an error, a list of alternatives was generated by the tool and suggested to the learner. Park et al. (2008) tested the tool in an interactive manner using five non-native speakers of English and focusing on the users' feedback about the tool rather than the system's intrinsic performance. The system was aimed at writing improvement rather than EDC, and "correctness" was treated as a relative rather than absolute value.

Yi et al. (2008) noted that the Web is a rich resource of examples of both correct and incorrect English writing. Their approach relied on the observation that learners often use the Web as a corpus of good English, feeding their queries into the search engines. If the phrase searched for has low or no counts on the Web, this is considered to be reliable evidence that the phrase is ungrammatical. However, since the Web can contain both grammatical and ungrammatical phrases, it is not the absolute count that matters but rather the difference between the counts for the alternatives: for example, Yi et al. (2008) report that **English as Second Language* has $306,000$ hits using the Google search engine, while *English as a Second Language* – $1,490,000$ hits. It shows that even if the count for the original phrase is high, the difference between the counts is more important. However, search engines are not able to correct the learners' original combinations if they are incorrect due to the wrong choice of content words, and this is where the error correction algorithm might be helpful.

Web counts were used in other research as well (Hermet et al., 2008; Tetreault and Chodorow, 2009). The approach is claimed to benefit from exposure to real language and a high number of examples, an opportunity to take language changes into account and to provide learners with snapshots of the original and corrected usage examples. However, there are a number drawbacks of using the Web for EDC, as the counts change from day to day, may differ from one search engine to another, represent the number of pages rather than the number of instances, and the number of queries per day is usually limited by search engines (see Kilgarriff (2007)).

The approach of Yi et al. (2008) did not rely on the use of resources or databases of correct and incorrect examples as other approaches did, and it also did not focus on any limited set of content word combinations specifically. They collected $1,012$ non-native sentences from ESL users' blogs, written mostly by Japanese, Korean and Chinese learners. The sentences were then checked and rewritten by a native speaker of English to produce more English-like alternatives. These rewrites were further used as a gold standard. Yi et al. (2008) focused on determiner errors and errors in VO and AN combinations. Their approach relied on query search of three types: *S-Query* for the whole sentence, *C-Query* for the chunks of related words within the sentence, and *W-Query* for individual keywords. The queries were searched for using the MSN search engine. First, the queries for the original sentences and combinations were searched, and if they were higher than a certain predefined threshold, the combination was considered to be correct. Otherwise, alternatives were searched for, this time using queries with all the same words in the context and the noun. The system tried to identify the most appropriate correction by comparing frequency. The authors reported the following results for the VO combinations: *recall* at

30.7%, *good precision*, or the proportion of times the system correctly identified an error and provided a correction that matched the native speaker's rewrite exactly, at 37.3%, *plausibly useful precision*, or the proportion of cases when the system correctly identified an error but the system's correction was different from the native speaker's edit, at 50%, and *false alarm* at 2.55%.

Östling and Knutsson (2009) focused on Swedish collocations including AN and VO combinations, but they did not use real learner data. Instead, they made a corpus of alternatives by artificially creating candidates with the original collocating words substituted with the semantically related words. As a result, they created a set of artificial miscollocations consisting of 60 examples. The algorithm was run on native Swedish text with a low expected error rate as well as on the set of artificial miscollocations, and each identified candidate collocation was compared to the alternatives with synonyms or otherwise related words substituted for the collocating word. The alternatives (synonyms) were searched for automatically using one of two methods: either in the synonym dictionary *Folkets Synonymlexikon*,[13] or via Random Indexing which returned a list of semantically related words including synonyms and antonyms. The collocational strength of the original and alternative suggestions was estimated using a combination of Mutual Information and Log Likelihood measures on the corpus consisting of Swedish Wikipedia, the Swedish PAROLE corpus[14] and Swedish newspaper articles. Östling and Knutsson (2009) reported that 85%-90% of the correct collocations from the native texts were left intact by the system, however, this result was partly due to the fact that for a high number of cases (25%) the system was not able to find an alternative. For 57% of the artificial miscollocations the tool found an acceptable alternative collocation, while in 5% of cases the acceptable suggestion was among the top three suggestions. However, since the method was applied to artificial examples created using a very similar technique to the one used for their correction, and since only semantically related confusions were considered, the results might be overestimated.

Liu et al. (2009) focused on error correction in the choice of verbs in VO collocations: 42 miscollocations from the list of 84 VO miscollocations listed in Liu (2002) were randomly selected to train the model, while the other 42 were used for testing. The model was trained using three types of features: the first type was based on word association estimated using mutual information for the suggested verb and the focal noun; the second feature type was based on semantic similarity between the correction candidate and the misused verb and was calculated on the basis of graph-theoretic distance between the WordNet synset containing the original verb and the synset containing the candidate; and the third type was based on the notion of the *collocation cluster* first introduced by Cowie and Howarth (1995): for example, *fulfil* and *reach* both collocate with *goal* which might make learners combine them in one collocation cluster. Since *fulfil* collocates with *ambition* and *purpose*, learners might assume that *reach* collocates with the same nouns which will result in such errors as **reach an ambition* and **reach a purpose*. Liu et al. (2009) generated collocation clusters for verbs that collocate with the focal nouns and nouns that the verbs collocate with. The possible verb substitutions were then chosen from among those which share most common collocations with the original verb from the collocation

[13] `http://folkets-lexikon.csc.kth.se/synlex.html`

[14] `http://spraakbanken.gu.se/eng/resource/parole`

cluster. The three types of features were combined using a Bayesian probabilistic model, and the k-best suggestions were returned for each miscollocation. The best results were obtained for a combination of all three feature types, while semantic similarity was reported to be the best-performing single feature. Precision was calculated for the list of the top k=[1, 10] suggestions, where a system's output was counted as correct if it matched any of the corrections provided by two annotators. System precision reported for the k=[1, 10] suggestions ranged from 53.57 to 94.05.

Wu et al. (2010) focused on VO miscollocations, and presented an online collocation writing assistant aimed specifically at academic writing and implemented using an ML classifier. Both training and test sets were collected from the *CiteSeer* database,[15] with the collocations extracted from the academic paper abstracts. In the experiment, the verbs were blanked out within the test sentences, and the task of the classifier was to predict the correct verb to be used, where the original verbs were assumed to represent the only correct candidate. A Maximum Entropy classifier was trained on 46, 255 examples representing the correct VO collocations with 790 verbs, using two types of features: a *head* or the focal noun, and *n-grams* of length 1 and 2 extracted from the surrounding contexts. The classifier learned the verb (class) to be predicted on the basis of the features, and for the 600 randomly selected test instances returned a list of verbs (classes) ordered by probability as an output. The best results were obtained using both types of features, with an MRR of 0.518, which means that the correct suggestion was ranked, on average, within the first two to three suggestions by the system. Since the approach originally used a single class as the gold standard, the reported results might be an underestimation of actual system performance. However, since the classifier was trained and tested on similar datasets this might have resulted in a limited set of possible collocations and verb classes, as well as reduced the sparsity of the feature set.

Finally, Dahlmeier and Ng (2011a) focused on miscollocations in Chinese and applied their approach to a test set of 856 sentences, extracted from the NUCLE, with one collocation error per sentence. Their algorithm was aimed at correction and assumed that the errors had already been detected by some ED algorithm. They assumed that a substantial portion of miscollocations can be traced back to L1.[16] But unlike Chang et al. (2008), they did not rely on manually constructed resources, and used an approach based on SMT. The use of SMT rather than reliance on resources helps overcome certain limitations of the latter, such as lower coverage. An advantage of the L1-based approach is that corrections found through spelling, homophones or synonyms help only with errors that stem from the L2, while translational equivalents help with errors that result from language interference. To support this idea, they compared performance of a baseline error correction system that used spelling variations, homophones and synonyms with a system that additionally used their SMT technique, and showed that the latter outperformed the former. They reported that the baseline system had an MRR of 0.08, while a combined system had an MRR of 0.17, with the SMT-based system alone performing with an MRR of 0.15.

Table 2.3 summarises the above-mentioned approaches, specifying the data used to extract the errors (*Data*), the type of the content word combinations (*Focus*), the data used

[15] `http://www.lib.utexas.edu/indexes/titles.php?id=84`

[16] They report that 1, 016 examples out of 2, 747 collocation errors extracted from the NUCLE data can be traced back to L1-transfer, and 906 of those cannot be explained by any other source of confusion.

Table 2.3: Summary of the previous approaches to EDC in content words.

Approach	**Data**	**Focus**	**Reference corpora**	**Detection/ Correction**	**Alternatives**	**Results**
Shei and Pain (2000)	Chinese speakers texts	VO	Correct (BNC) & miscollocations	Both (lookup, comparison)	Synonyms (WordNet)	Not reported
Wible *et al.* (2003)	EnglishTLC	Various, incl. VO, AN	Correct & corrected (EnglishTLC)	Both (lookup)	Corrections stored	P= 0.96
Chang *et al.* (2008)	Chinese miscollocations	VO	Correct (BNC) & errors stored	Both (lookup, comparison)	L1-related (bilingual dictionary)	$Acc_{detect.} = 0.94$ $P_{correct.} = 0.84; 0.94$[a] $MRR_{correct.} = 0.50; 0.66$
Futagi *et al.* (2008)	TOEFL essays	Various, incl. VO, AN	Correct (Lexile & SourceFinder)	Both (lookup, comparison)	Synonyms (WordNet, Roget's Thesaurus)	$F_{correct} = 0.83$-0.84 $F_{misused} = 0.31$-0.34
Park *et al.* (2008)	Interactive tool	Various	*n*-gram dictionary from corpus	Both (lookup, comparison)	Synonyms (WordNet) & other operations	Not reported
Yi *et al.* (2008)	Asian ESL learners' blogs	VO	Web	Both (Web counts)	Used in same snippets (Web)	$R = 0.31$ $P = 0.37$
Östling and Knutsson (2009)	Swedish collocations	Various, incl. VO, AN	Correct (Swedish corpora)	Both (comparison)	Synonyms (dictionary) & Random Indexing	$P_{correct} = 0.85$-0.90 $P_{misused} = 0.57$
Liu *et al.* (2009)	Chinese miscollocations	VO	–	Correction (prob. model)	Similar (WordNet) & colloc. clusters	$P = 0.54$-0.94
Wu *et al.* (2010)	CiteSeer abstracts	VO	–	Both (machine learning)	Chosen by a machine classifier	$MRR = 0.52$
Dahlmeier and Ng (2011)	NUCLE	Various	–	Correction (SMT)	Spelling (edit dist.), homophones (CuVPlus), synonyms (WordNet) & SMT	$MRR_{machine} = 0.17$ $MRR_{\cap(human)} = 0.33$ $MRR_{\cup(human)} = 0.57$

[a] The second value reported for P and MRR after the semicolon is the improved result after manual qualitative evaluation.

to assess the correctness of the word combinations (*Reference corpora*), whether these approaches addressed error detection, correction, or both (*Detection/Correction*), how the possible corrections were found (*Alternatives*), and finally, the results that were reported (*Results*). Table 2.3 shows that the approaches used different datasets and evaluation techniques which complicates direct comparison of the results.

2.4 Semantic models

One important aspect that the models based on superficial collocational strength of the content word combinations do not capture is the meaning of the individual words chosen and the way the meaning of their composition is derived. Semantic mismatch between the chosen words is a rich source of errors in learner writing. Learners lacking deeper understanding of how words in L2 should be combined often inadvertently produce semantically deviant combinations. Obviously, producing nonsensical content word combinations is not what the learners aim for, therefore, the semantic mismatch is a property of the resulting combination rather than the original source of the error.

For instance, learners are often confused by words that sound or are spelled similarly. This is often aggravated by the fact that certain morphologically related words are also similar in meaning: frequent confusion pairs of this type include *economic* versus *economical*, *historic* versus *historical*, *old/older* versus *elder*. Although the adjectives share certain semantic components, they also differ with respect to certain semantic properties. If a learner produces an AN **elder neighbour*, a native speaker can still understand the original meaning by backtracking the process of word selection, but can also see that, from a semantic point of view, the combination is deviant. The adjective *elder* has a semantic component of age comparison and, to a certain degree, can express the meaning intended. However, it also has a very specific meaning of referring to one's family, and in combination with any term not denoting a family member represents a semantic mismatch.

The reasons for incorrect word choice might be diverse (see §2.3.2). We note that independent of the reasons, the resulting combinations might display semantic mismatch of various degrees. We believe that the approaches that can model the meaning of the words and their combination are a powerful means of detecting incorrect word choice.

We use models of distributional and compositional semantics, with the former being used to represent the meaning of the words, and the latter being used to model the meaning of the content word combinations. We also note that the closest phenomenon to learner errors in the choice of content words that can occur in native writing is semantic anomaly, and our methods are inspired by the previous work on semantic anomaly detection using compositional distributional semantic models by Vecchi et al. (2011).

2.4.1 Distributional semantic models

The underlying assumption of the distributional approach is expressed with the *distributional hypothesis* which states that a word is characterized by "the company it keeps" (Firth, 1957, p. 11). Sahlgren (2008) quotes other relevant definitions, including “words which are similar in meaning occur in similar contexts” (Rubenstein and Goodenough, 1965); “words with similar meanings will occur with similar neighbors if enough text material is available” (Schütze and Pedersen, 1995); “a representation that captures much of how words are used in natural context will capture much of what we mean by meaning” (Landauer and Dumais, 1997); and “words that occur in the same contexts tend to have similar meanings” (Pantel, 2005). In essence, the distributional hypothesis claims that the words that co-occur with the target word and the contexts in which the target word occurs implicitly describe the meaning of the word. Therefore, the meaning of the word can be “accessed” through the observed examples of the word’s use in context and represented with its co-occurrence counts with the other words in its context.

It is believed that the meaning of a word can be sufficiently described via the word’s distributional pattern. One of the most widely accepted notions of *distribution* is derived from Harris (1954) and refers to a methodology by which it is supposed to be possible to identify linguistic units of all sizes by observing the contexts in which such units occur, by a process of formal comparison and contrast (Pulman, 2013, p. 336). Distributional methodology, based on such formal comparison and contrast, makes it possible to identify word’s meaning not via its reference to the extra-linguistic information, but rather via the differences in the distribution of words. As Harris (1954, p. 43) notes, “if we consider words or morphemes A and B to be more different in meaning than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C. In other words, difference of meaning correlates with difference of distribution”. From that, it follows that the words with similar distributions have similar linguistic meaning: “it is possible to measure how similar are the selection approximations of any two words” (Harris, 1954) and “if A and B have almost identical environments except chiefly for sentences which contain both, we say they are synonyms” (Harris, 1954).

Sahlgren (2008) points out that “we should be able to populate a distributional model with syntagmatic relations if we collect information about which words tend to co-occur, and with paradigmatic relations if we collect information about which words tend to share neighbours”. Syntagmatic relations are covered by distributional patterns for the target words, while paradigmatic relations between target words are accessible through comparison and identification of similar distributional patterns. Sahlgren (2008) gives a definition for the *refined distributional hypothesis*: “A distributional model accumulated from co-occurrence information contains syntagmatic relations between words, while a distributional model accumulated from information about shared neighbours contains paradigmatic relations between words”.

The most obvious case of word co-occurrence (syntagmatic relation) is a collocation, though words co-occurring within one clause, sentence, paragraph or even document also stand in syntagmatic relations. The size of the context window to consider, as well as the type of the context, is one of the parameters of the distributional models to be set: for example, one could consider n-word context windows, sentence-internal co-occurrence (Baroni and Zamparelli, 2010), or dependency-based contexts (Padó and Lapata, 2007). The particular choice depends on the type and amount of information required for the task. Smaller context windows are able to provide more relevant semantic information, but will inevitably lead to very sparse and less statistically reliable representations. Some relevant words can also be found outside of the n-word context window or words grammatically related to the target one, but will also contain irrelevant information since, as Ruge (1992) notes, "in large contexts nearly every term can co-occur with every other; thus this must not mean anything for their semantic properties". Sahlgren (2008) notes that a clause or a sentence, being linguistic universals existing in every language, are most suitable for collecting context information.

The most straightforward mathematical model for representing word meaning through its distribution is a vector. *Distributional semantic models* (*DSM*) use n-dimensional vectors, where the number of dimensions is defined by the number of content words with which the target word co-occurs within the chosen context window. *Semantic space* is the collection of the target words represented with the n-dimensional vectors. It is usually represented with a *co-occurrence matrix* where rows are the distributional vectors for the target words and columns represent the contextual words.

Let us consider a sample semantic space containing vectors for the target words *elder*, *neighbour*, *brother* and *sister* from the example cited above. Let us assume that the set of context words contains, among others, nouns *mother* and *street*, adjectives *older* and *noisy*, and verbs *lose* and *move*. The vectors will contain the sentence-internal co-occurrence counts for the target words and the context words. Table 2.4 gives an example of the semantic space with the co-occurrence counts for the selected context words estimated using the BNC. For the sake of this example, we will only focus on these 6 dimensions of the target word vectors, using '...' to represent the other dimensions of the vector.

Words	*mother_n*	*street_n*	...	*older_a*	*noisy_a*	...	*lose_v*	*move_v*	...
elder_a	45	10	...	13	0	...	7	7	...
neighbour_n	68	91	...	12	21	...	36	61	...
brother_n	424	113	...	233	3	...	112	93	...
sister_n	567	52	...	168	3	...	82	75	...

Table 2.4: An example of the semantic space for the chosen target words.

From the distributional patterns represented by these 6 selected dimensions, we can deduce that it is more typical to associate *brothers* and *sisters* with a *mother* than with a *street*, while one speaks about *neighbours* in the context of a *street* more often than in the context of a *mother*. It is also more relevant to talk about *older brothers* and *sisters*, while the concept of *noisiness* is more relevant for *neighbours*. One can also *lose* a *brother* or a *sister* but less often a *neighbour*, while with respect to *neighbours* it is more typical to talk about *moving*.[17]

We can note that in our example the distributional patterns for *brother* and *sister* are quite similar, while being different from that for *neighbour*. The distributional pattern of *elder* is also more similar to those of *brother* and *sister* than to that of *neighbour*. This illustrates an important property of distributional semantic models: due to the fact that semantically similar words occur in similar contexts, they are situated in the semantic space closer to each other than to the words which are less similar. In practice, this can be estimated using the *cosine* measure which is a function of the width of the angle formed by two vectors:

[17] We should admit that this is a very simplistic example, since we are only considering a specifically selected set of vector dimensions. Our goal here is to illustrate, in a simple form, how semantic similarity can be estimated on the basis of the distributional patterns.

$$Similarity = cos(\theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{2.6}$$

The *cosine* measure is one of the most widely used measures for estimating vector similarity, providing a clear interpretation for similarity in the semantic space. For a discussion of other measures of semantic similarity, see Kiela and Clark (2014).

Table 2.5 presents the similarity values for the set of target words in our example which are estimated using the subset of the co-occurrence counts with the chosen context words. Even though in a real situation we will be dealing with a higher number of dimensions and use a bigger number of co-occurrence counts to estimate similarity than just the 6 as for this example, the obtained results give a rough idea of how we can measure semantic similarity between words using DSMs. We see that the distance values support our linguistic intuition that *brother*, *sister* and *elder* are all similar to each other, while *neighbour* is more distant.

Words	*elder_a*	*neighbour_n*	*brother_n*	*sister_n*
elder_a	1.0000	0.7222	0.9745	0.9925
neighbour_n		1.0000	0.7338	0.6438
brother_n			1.0000	0.9645
sister_n				1.0000

Table 2.5: Similarity between the chosen target words.

There is a direct geometric interpretation for the DSMs: since we are representing the words as vectors in the semantic space, we can also visualise them as geometric objects in the n-dimensional space. For example, if we focus on the first two dimensions in our space, we can represent the target word vectors as in Figure 2.1.

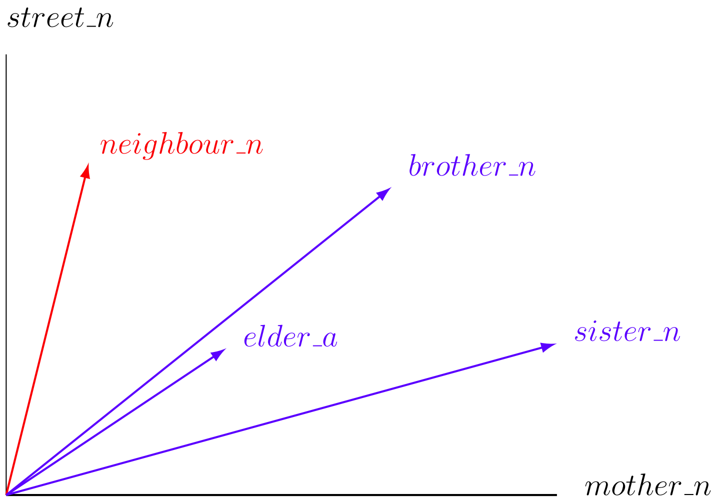


Figure 2.1: Semantic space example.

We see that the vectors for *elder*, *brother* and *sister* are situated closer together and also closer to *mother*, while the vector for *neighbour* is placed further apart from other vectors and closer to *street*.

The vectors that are placed close together in the semantic space are called neighbours. They exemplify what Sahlgren (2008) refers to as paradigmatic relations – relations that hold between words or linguistic entities that occur in the same context but not at the same time. The syntagmatic and paradigmatic relations are not mutually exclusive: in our example, *brother* and *sister* would stand in both relations since they can occur in the same context as well as in similar contexts. Paradigmatic relations between *elder* and *older* would be more pronounced than syntagmatic, since the two words would typically occur in a similar rather than the same context. The strength of DSMs is that they can account for word meaning and give access to the related words.

There are a number of other implementation issues related to DSMs. Sahlgren (2008) mentions that some linguistic information can be added to the models by lemmatising words and disambiguating them with PoS tags. Certain modifications of the semantic space can lead to collecting more relevant information at the cost of data-sparseness or to more general information at the cost of losing some fine-grained distinctions.

The frequency of individual words plays a role in the magnitude of the absolute counts: for example, it makes sense to compare the patterns for *elder* and *brother* but not the absolute counts as the noun *brother* is more frequent (occurs $11,281$ times in the BNC) than the adjective *elder* (occurs $1,058$ times). The co-occurrence counts are influenced by both the frequency of the target word and the frequency of the context words. Therefore, in practice, the counts are weighted by the frequency of both words, for example, using $TF\text{-}IDF$ or *mutual information*. Another effect of applying a weighting scheme to the raw co-occurrence counts is that it shows the association of the target and context words discounting the weights of components associated with contexts with high probability of chance occurrence (Evert, 2005). In spite of the previous work that discusses parameter settings for DSMs (Kiela and Clark, 2014; Lapesa and Evert, 2013; Bullinaria and Levy, 2007, 2012), there is no single best setting for a semantic space and the particular implementation details depend on the task.

The appropriateness of using DSMs to represent word meaning has been tested in a number of areas from linguistics to cognitive science to neuroscience. It has been shown that these models are successful in simulating many aspects of human semantic performance, mimic the process of language acquisition, and are good predictors of the patterns of brain activation recorded in subjects thinking of a concept (Baroni et al., 2014a; Murphy et al., 2012; Lenci, 2008; Mitchell and Lapata, 2008; McDonald and Ramscar, 2001). Baroni et al. (2014a, p. 21) conclude that "a core aspect of the meaning of a word is given by (a function of) its distribution over the linguistic contexts [...] in which it occurs, encoded in a vector of real values that constitutes a feature-based semantic representation of the word".

The question then is whether these models can be applied to a linguistic entity of arbitrary length. From the cognitive point of view, we are exposed to a huge number of words throughout our lives and acquire their meaning partly through their distribution. Experiments by McDonald and Ramscar (2001) confirm that human subjects change their idea of a rare or nonce word meaning depending on the distributional properties of the contexts in which these words are presented. Data sparsity is a natural property of human language, and we can encounter some words very often so that we form a reliable meaning representation for these linguistic entities, but we might never encounter all the possible combinations with all the words we know. Yet, we are still able to derive the meaning of a longer linguistic expression if we know the meaning of its components. This suggests that DSMs are less appropriate for meaning representation of linguistic units beyond words: from the computational point of view, they cannot be used to derive statistically reliable representations for longer phrases. For example, let us consider the distributional vectors using the same contextual elements for the ANs *elder neighbour*, *elder brother* and *elder sister*. The vectors are presented in Table 2.6.

Words	*mother_n*	*street_n*	...	*older_a*	*noisy_a*	...	*lose_v*	*move_v*	...
elder neighbour	0	0	...	0	0	...	0	0	...
elder brother	11	3	...	5	0	...	2	3	...
elder sister	7	0	...	0	0	...	0	0	...

Table 2.6: Semantic space for AN combinations.

Since *elder neighbour* is not attested in the BNC, all of its co-occurrence counts are 0. However, we see that the counts for the other two ANs are very low, too: they are much lower than the counts for the individual words within these combinations, and of the 6 dimensions only one is filled for *elder sister*. This shows that the distributional models for the longer combinations are usually very sparse and less statistically representative even for relatively frequent word combinations.

Absence of a word combination in a corpus of English can signal that the combination is non-sensical, semantically deviant, or plainly incorrect. In fact, *elder neighbour* is annotated as an error in our learner error dataset. At the same time, language is productive and the number of acceptable content word combinations is bigger than what any corpus can cover reliably. A bigger corpus can provide us with more reliable distributional patterns for the words and their combinations, but the problem of data sparsity will inevitably occur, especially when the length of the combination considered increases.

Compositional distributional semantic models provide a better means of modelling content word combinations than using their corpus occurrences.

2.4.2 Models of semantic composition

While we rely on distributional models to represent the meaning of words, the meaning of longer linguistic entities is handled by formal semantics, which in the Fregean tradition (Frege, 1892; Montague, 1970, 1973) relies on the principle of *compositionality* which states that the meaning of a complex expression is determined by its structure and by the meanings of its constituents, or that the meaning of a complex expression is some function of the meanings of its components (Baroni et al., 2014a; Pulman, 2013, p. 334). For example, the meaning of *elder sister* is a function of the meaning of the noun *sister* and its modifier *elder*.

From the computational point of view, this amounts to applying a certain function to the distributional vectors representing the constituent words of a phrase. Baroni et al. (2014a) review a number of models of semantic composition and functions applied to the distributional vectors within these models. We focus on three of them – the simple *additive* (*add*) and *multiplicative* (*mult*) models by Mitchell and Lapata (2008, 2010), and the *adjective-specific linear maps* (*alm*) by Baroni and Zamparelli (2010).

Composition by component-wise operations

These models are also referred to as *component mixture* models (Baroni et al., 2014a). In the general form, components of the composite vector c are derived by application of a certain mathematical operation $\diamond$ to the components of the input vectors a and b:

$$c_i = a_i \diamond b_i \tag{2.7}$$

The most widely used models of this type are the *add* and *mult* models of Mitchell and Lapata (2008, 2010) which use component-wise addition $+$ and multiplication $\odot$ to derive the composite vectors. Baroni et al. (2014a, p. 27) also note that the *additive* approach was the most common one in composition in distributional semantics from early on (Kintsch, 2001; Foltz et al., 1998; Landauer and Dumais, 1997).

Table 2.7 shows how the composite AN vectors are derived from the distributional vectors presented in Table 2.4. The modelled vectors are much less sparse than the AN distributional vectors extracted from the same corpus (see Table 2.6). At the same time, the distributional pattern is preserved, and we can see that the *mother*, *older* and *lose* dimensions are more pronounced for *elder brother* and *elder sister* with both models.

ANs	*mother_n*	*street_n*	...	*older_a*	*noisy_a*	...	*lose_v*	*move_v*	...
elder + *neighbour*	113	101	...	25	21	...	43	68	...
elder $\odot$ *neighbour*	3060	910	...	156	0	...	252	427	...
elder + *brother*	469	123	...	246	3	...	119	100	...
elder $\odot$ *brother*	19080	1130	...	3029	0	...	784	651	...
elder + *sister*	612	62	...	181	3	...	89	82	...
elder $\odot$ *sister*	25515	520	...	2184	0	...	574	525	...

Table 2.7: Derivation of the AN vectors with the *add* $+$ and *mult* $\odot$ models.

Mitchell and Lapata (2008, 2010) characterise the interaction of distributional properties of the *mult* model as a quantitative form of "feature intersection". By analogy, the *add* model can be said to represent feature union, as the modelled vectors inherit cumulative score mass from the corresponding input components and can inherit a high value in a particular component from one of the input vectors. For example, high cumulative counts are inherited by *elder* + *brother* and *elder* + *sister* in the *mother*-related dimension from the input noun vectors rather than the adjective vector. The effect of the high input vector counts is even more pronounced within the *mult* model. However, since the *mult* model captures feature interaction, zero counts in the input vectors result in zero counts in the composite vector showing that, for example, *noisy* is not relevant for *elder brother* and *elder sister* since it is not relevant for *elder*.

These models, in their simplest form, are also easy to implement in practice since they require no additional training or parameter tuning. They have been, so far, most widely used in compositional distributional semantics and successfully applied to a number of tasks (Foltz et al., 1998; Erk and Padó, 2008; Mitchell and Lapata, 2009; Grefenstette and Sadrzadeh, 2011; Vecchi et al., 2011; Blacoe and Lapata, 2012; Boleda et al., 2012).

However, these models have a serious deficiency which might make them inappropriate for some linguistic tasks: both models derive the composite vectors in a symmetric way with both input vectors contributing equally to the combination. They are referred to as *mixture* models because they essentially "mix" distributional counts from the input vectors. From the linguistic point of view, adjectival modification is an asymmetric operation with the input noun (head of the combination) contributing to the resulting composition more than the adjective. For example, *elder brother* inherits some properties from *elder* which distinguish it from simply a *brother*, but our linguistic intuition would suggest that it still inherits much more from *brother*: an *elder brother*, while being both an *elder* family member and a *brother*, is still to a greater extent a *brother*. The simple *add* and *mult* models have no way to take the grammatical relations between the words into account. One practical solution for avoiding this is to introduce weights and weigh the head of the combination heavier than the modifier: Mitchell and Lapata (2010) discuss a weighted additive model within which the modifier vector can, for example, be multiplied by 0.2 and the head vector by 0.8. In this case, however, the model might require some parameter tuning.

The simple *add* and *mult* models might not be suitable for the linguistic tasks where the grammatical structure of the word combination is important, as they would fail to distinguish between *dog trainer* and *trainer dog*, or *dog chase cat* and *cat chase dog* (Baroni et al., 2014a, p. 28). However, they can still be used in a number of tasks where other aspects of meaning composition are considered, for example, in detecting semantic anomaly.

Distributional functions and linear transformations

Baroni and Zamparelli (2010) and Baroni et al. (2014a) propose a different type of compositional distributional semantics model. They note that a mixture of the vector components will not always provide a satisfactory representation: while one can say that *old cat* is a mixture of the features of *old* things and *cats*, a similar interpretation for *some cats* – as a mixture of *some* things and *cats* – is much less appealing. While certain words are defined more directly by their distribution (for example, a noun *cat*), some other words might "adjust" their meaning depending on the words they modify. Consider an example of the adjective *brown*: "In order for a cow to be brown most of its body's surface should be brown, though not its udders, eyes, or internal organs. A brown crystal, on the other hand, needs to be brown both inside and outside. A book is brown if its cover, but not necessarily its inner pages, are mostly brown, while a newspaper is brown only if all its pages are brown. For a potato to be brown it needs to be brown only outside..." (Lahav, 1993, p. 76). Interpretation of many verbs also depends on their arguments: for example, the sense of *take* in *take a note*, *take a class* and *take a shower* is not the same.

Baroni and Zamparelli (2010) and Baroni et al. (2014a) propose a type of model capable of capturing these phenomena: they suggest representing certain linguistic categories, for example nouns, with distributional vectors, while modelling other categories, for example, adjectives, verbs, determiners, prepositions and so forth, by *distributional functions*. This approach allows us to speak of different compositional structures derived through function application: $\mathcal{OLD}(\mathbf{cat})$, $\mathcal{SOME}(\mathbf{cat})$ and $\mathcal{TAKE}(\mathbf{note})$. Baroni et al. (2014a, p. 33) note that this approach is closer to the classic, formal semantics treatment of compositionality.

The algebraic equivalent to the compositional functions in the vector-based distributional framework are *linear transformations* or *linear maps*. A linear transformation takes a vector of size J and returns a vector of size I (J might equal I), where each output component is a linear combination of all input components, or a weighted sum of the input components (Baroni et al., 2014a, p. 33). The linear transformation can be encoded with a matrix: given the J- and I-dimensional vector spaces, any linear transformation from the first onto the second is entirely characterised by a matrix of shape $I \times J$, and the application of the linear transformation to the input vector is given by the product of the matrix by the vector from the J-dimensional space.

For example, if we have a matrix $\mathbf{M}$ of dimensionality $I \times J$, which encodes the linear transformation, and an input vector $\mathbf{v}$ of size J, the matrix-by-vector multiplication will result in the I-sized vector $\mathbf{w} = \mathbf{M} \times \mathbf{v}$, with each component of this vector defined by:

$$w_i = \sum_{j=1}^{j=J} M_{ij} \times v_j \tag{2.8}$$

The component w_i of the resulting vector is, thus, a weighted sum of all J components of the input vector, each multiplied by the value in the ij-th cell of the matrix.

Baroni *et al.* suggest treating distributional functions as linear transformations on semantic space, where first-order one-argument distributional functions such as adjectives or intransitive verbs can be encoded with matrices. Higher-order functions such as transitive verbs can be encoded with *tensors* of the appropriate dimensionality – more complex algebraic structures which extend matrix representations to higher number of dimensions (Grefenstette, 2013).

The matrix-by-vector multiplication for the first-order functions can be represented as:

$$f(a) =_{def} \mathbf{F} \times \mathbf{a} = \mathbf{b} \tag{2.9}$$

where $\mathbf{F}$ is the matrix encoding the function f as a linear transformation, $\mathbf{a}$ is the vector for the argument a and $\mathbf{b}$ is the vector output to the composition (Baroni et al., 2014a).

For example, let us assume that the linear transformation representing an adjective *elder* is encoded with a matrix $\mathbf{M}$ which, for the sake of example, is reduced to only two dimensions for *lose_v* and *move_v*:

$$\mathbf{M} = \begin{array}{c c} & \begin{array}{cc} \textit{lose\_v} & \textit{move\_v} \end{array} \\ \begin{array}{c} \textit{lose\_v} \\ \textit{move\_v} \end{array} & \begin{pmatrix} 10 & 3 \\ 0 & 7 \end{pmatrix} \end{array}$$

and the noun *brother* is represented with a corresponding 2-dimensional vector $\mathbf{v}$:

$$\mathbf{v} = \begin{array}{c c} \begin{array}{c} \textit{lose\_v} \\ \textit{move\_v} \end{array} & \begin{pmatrix} 112 \\ 93 \end{pmatrix} \end{array}$$

Then the matrix-by-vector multiplication will result in the vector $\mathbf{w}$ for $\mathbf{ELDER}(\mathbf{brother})$:

$$\mathbf{w} = \mathbf{M} \times \mathbf{v} = \begin{pmatrix} 10 & 3 \\ 0 & 7 \end{pmatrix} \times \begin{pmatrix} 112 \\ 93 \end{pmatrix} = \begin{pmatrix} 10 \times 112 + 3 \times 93 \\ 0 \times 112 + 7 \times 93 \end{pmatrix} = \begin{pmatrix} 1399 \\ 651 \end{pmatrix}$$

The matrix illustrates the contribution of the different input components to the output vector: the ij-th cell contains the quantity determining how much the component corresponding to the j-th input context element contributes to the value assigned to the i-th context element in the output vector. For example, matrix **M** encoding the adjective *elder* contains 0 in the cell corresponding to the input context label *lose_v* and the output context label *move_v*. This means that the *lose*-labelled component of the input noun vector will contribute 0 value to the *move*-labelled component of the *elder N* output vector.

The linear transformation-based approach has a number of properties that simpler models lack: they do not simply mix the distributional counts of the input vectors, clearly taking into account the interaction between different dimensions of the input structures; they are not symmetric and take the difference in the function of the words into account; they can be applied recursively in a syntax-aware manner and combining different types of semantic representations for longer linguistic entities.

From the practical point of view, the question is how to induce these higher-order objects from the data in a similar way as we derive vectors from the word's distribution, and how to determine the values to fill the cells of the tensors. In accordance with the distributional semantics tradition, we can again look at how the words and word combinations are used in the data and induce the weights determining the contribution of the components of the input vectors to the output vectors. The idea is that, to learn the distributional function for the adjective *elder*, we can collect some *training* example pairs like <*brother, elder brother*>, <*child, elder child*>, <*daughter, elder daughter*> and so forth from the corpus. Then, using these input–output pairs and applying regression, we can find the weights that, on average, provide the best approximation to each output component as a weighted sum of the corresponding input components across the training set. The learning algorithm guarantees that the set of weights in the derived matrix produces the best approximations to the corpus-extracted output vectors when multiplied by the corresponding corpus-extracted input vectors (Baroni et al., 2014a, p. 46).

Baroni and Zamparelli (2010) have presented qualitative evidence that AN vectors extracted from the corpus in general make intuitive sense by showing that they are surrounded by semantically similar words and phrases. Additional evidence for the fact that corpus-harvested distributional phrase vectors are high-quality examples of the composite meaning that they represent is provided by Boleda et al. (2012) and Turney (2012). Therefore, the corpus-observed phrase vectors (for example, AN vectors) can be used as targets for learning the distributional functions (for example, adjectives) from the mappings of the observed words (nouns) and the correspondent word combinations (ANs). Baroni et al. (2014a) also note that since the distributional adjective function is trained on numerous examples, its application to input vectors produces a better approximation to the meaning of the phrase than many corpus-derived distributional phrase vectors, especially for rare word combinations. Application of the distributional function for *elder* to the noun vector of *sister* in our case might produce a better approximation of the meaning of *elder sister* than the distributional vector (see Table 2.6).

Baroni and Zamparelli (2010) presented the *alm* model in which adjectives are represented by matrices and nouns are represented by distributional vectors. A matrix for each adjective adj_i is learned from the corpus-attested pairs of <*noun, adj_i–noun combination*> vectors using partial least squares regression. We follow this experimental setting.

2.4.3 Semantic anomaly detection

Detection of semantic anomaly in AN combinations is one of the tasks on which compositional distributional models have been tested (Vecchi et al., 2011). We review this work here because, first of all, semantic anomaly in native English is the closest phenomenon to that of content word errors in learner English: since native speakers are assumed to be experts in their own language (see §1.1), it can be expected that they only produce semantically anomalous combinations on purpose. In contrast, non-native speakers may lack the ability to choose content words appropriately and incorrect word choice can result in semantically anomalous combinations. If compositional distributional models can be applied to detect semantic anomaly, they can also be applied to detect errors in content words. Secondly, we believe that this task tests the crucial ability of the semantic models to account for compositionality in language. Whereas approaches based on corpus evidence can only deal with the linguistic phenomena that have been seen before, compositional distributional semantic models can help us derive the meaning of the longer linguistic units from the meaning of their parts.

Vecchi et al. (2011) have studied a set of corpus-unattested AN combinations. Absence of a combination in a corpus of English, however big this corpus might be, can suggest that the combination is semantically anomalous, but it can also be due to a variety of different reasons including pure chance, the fact that the expression, though understandable, is ungrammatical, that it uses a rare or very complex structure, describes false facts or nonexistent entities (Vecchi, 2013). Generally, it also shows that a corpus of any size cannot cover all possible semantically acceptable word combinations because natural language is productive (Chomsky, 1957). Vecchi et al. (2011) have shown that it is possible to use compositional distributional methods to distinguish *unattestedness due to nonsensicality* from *unattestedness due to other reasons*.

We believe that experiments on datasets of corpus-unattested examples test the ability of compositional distributional semantic models under the extreme circumstances when modelling the meaning of the combination is the only way to derive it. Since the meaning of the complex expression is modelled from the meaning of its parts, the size of corpus against which the attestedness of the AN combinations is checked is of less importance.

Experimental setting

The coverage of any corpus as well as distributional models which rely on corpus evidence are limited. However, even if we have never seen a *blue rose*, the concept itself is not inconceivable, while that of a *residential steak* is much less easy to interpret (Vecchi et al., 2011). Both might be unattested in a corpus of English, but the reason for the absence of the former is data sparsity, while the latter is absent due to its semantic anomaly. Vecchi et al. (2011) collected and presented a set of corpus-unattested AN combinations that we described in §2.1.4.

The semantic space for the experiments was constructed using as the target word vocabulary the $8K$ most frequent nouns and $4K$ most frequent adjectives from the concatenated corpus of native English consisting of the ukWaC, the English Wikipedia and the BNC. For both nouns and adjectives, the top 50 most frequent elements were excluded as they might have too broad a meaning and the ability to combine with almost any noun or adjective, which will make the test examples less challenging. The semantic space was also populated with the vectors for $33K$ ANs which resulted in a total of $45K$ target elements in the semantic space.

The ANs were generated by first crossing a selected set of 200 very frequent adjectives attested in the corpus at least $47K$ times, and at most $740K$ times, and the set of the $8K$ nouns from the semantic space vocabulary. Among these generated ANs, a set of $30K$ ANs attested in the corpus at least 200 times was randomly sampled, and to add further variety to the semantic space, augmented by a less controlled second set of $3K$ ANs randomly picked among those that were attested and were formed by the combination of any of the $4K$ adjectives and $8K$ nouns in the target word vocabulary. The set of generated but corpus-unattested AN combinations was used to generate the AN dataset.

The semantic space was constructed using 10K-dimensional vectors for the vocabulary of $45K$ target elements. The vectors encoded sentence-internal co-occurrence of the target elements with the top $10K$ most frequent content words in the concatenated corpus. The raw co-occurrence counts were then transformed into *Local Mutual Information* (*LMI*) scores (Baroni and Lenci, 2010; Evert, 2005). This weighting scheme was chosen as it closely approximates the commonly used Log-Likelihood Ratio while being simpler to compute. The full co-occurrence matrix was then reduced using *Singular Value Decomposition* (*SVD*) (Landauer and Dumais, 1997; Rapp, 2003; Schütze, 1997) to yield a smaller and denser space which is easier to use with computationally intensive models like linear transformations. The original $45K$-by-$10K$-dimensional matrix was reduced to a $45K$-by-300 matrix, where vocabulary items were represented by their coordinates in the space spanned by the first 300 right singular vectors of the SVD solution.

Four compositional methods including the simple *add* and *mult* models of Mitchell and Lapata (2008, 2010) and the *alm* model of Baroni and Zamparelli (2010) were used. The fourth model, the *linear map (lm)* approach proposed by Guevara (2010), derives the composite AN vector by multiplying a weight matrix by the concatenation of the adjective and noun vectors, so that each dimension of the AN vector is a linear combination of dimensions of the input adjective and noun vectors, which distinguishes it from the *alm* approach of Baroni and Zamparelli (2010). The matrix coefficients for the *alm* and *lm* models were estimated using multivariate *partial least squares regression* (*PLSR*) as implemented in the *R pls* package (Mevik and Wehrens, 2007), with the latent dimension parameter set to 50. The number of training examples per adjective for the *alm* model ranged from 100 to more than 500 items depending on the available training data.

Measures of semantic anomaly

Vecchi et al. (2011) proposed looking at three properties of the model-generated vectors that could distinguish between the representations of the semantically acceptable and semantically deviant AN combinations:

1. **Vector length**: they hypothesised that, since the values in the dimensions of a semantic space are a distributional proxy to the meaning of an expression and since a semantically deviant word combination is formed by combination of semantically incompatible words, the distributional counts in the input vectors will be distributed along different dimensions and the resulting vector will in general have low values across the dimensions. For example, a *parliamentary potato* is no longer a vegetable but it is also unlikely that it denotes an entity related to parliament. Therefore, it might have low values in both dimensions characterising vegetables and dimensions characterising parliament-related entities. The first measure proposed is the length of the model-generated AN vector, and the hypothesis is that vectors for semantically anomalous ANs are shorter than those for semantically acceptable ANs.
   In our set of examples:
   $len(elder{+}neighbour)=174.67 < len(elder{+}brother)=565.49$;
   $len(elder{\odot}neighbour)=3234.48 < len(elder{\odot}brother)=19378.77$.[18]

2. **Similarity/Distance to the input noun**: the second hypothesis is based on the observation that anomalous composition destroys or randomises the meaning of the input noun, and as a side effect one might expect the resulting AN to be more distant, in the semantic space, from the component noun. For example, although a *marble iPad* might have lost some essential properties of iPads, it must still retain at least some characteristics of iPads, for example, being shaped like one (Vecchi et al., 2011). Still, we cannot imagine what a *parliamentary potato* should be and cannot attribute even a subset of regular potato properties to it. We can measure the similarity between the model-generated AN vector and the input noun vector with the cosine of the angle between two vectors, and it is expected to be lower for the semantically anomalous combinations.
   In our set of examples:
   $cos(neighbour,(elder{+}neighbour))=0.9811 < cos(brother,(elder{+}brother))=0.9998$;
   $cos(neighbour,(elder{\odot}neighbour))=0.7452 < cos(brother,(elder{\odot}brother))=0.9042$.
   The difference for the *add* model representation might not be significant, but that for the *mult* model is more evident.

3. **Density of the neighbourhood**: the third hypothesis is based on the assumption that if an AN makes no sense, its model-generated vector should not have many neighbours, since the semantic space is populated by nouns, adjectives and ANs that are commonly encountered in the corpus and should all be meaningful. It is expected that semantically anomalous ANs will be "semantically isolated". This is measured by calculating the average cosine with the top 10 nearest neighbours, and the model-generated vectors of semantically anomalous ANs are expected to have lower density than model-generated acceptable ANs.
   In our set of examples:
   $dens(elder{+}neighbour)=0.9518 < dens(elder{+}brother)=0.9947$;
   $dens(elder{\odot}neighbour)=0.9496 < dens(elder{\odot}brother)=0.9886$.

[18] We should note that since in these examples we use raw rather than normalised counts the difference in the length might be a consequence of the difference in the length of the input vectors.

Evaluation and results

Vecchi et al. (2011) applied the measures of semantic deviance to the two groups of model-generated vectors – vectors representing the acceptable and the anomalous combinations. The two sets of vectors were compared, for each composition method and measure, by means of two-tailed Welch's t tests. The estimated t score, which shows the standardised difference between the mean acceptable and anomalous AN values, was reported.

Their results show that the *add* and *mult* models, in spite of the simple idea they are based upon, provide significant results for 2 out of 3 measures – for vector length and density, only failing the cosine test. The model of Guevara (2010) does not distinguish between the two groups of vectors reliably with any of the measures, while the *alm* model captures the distinction in terms of density. Overall, the density measure performed most reliably of all proposed measures, showing statistically significant differences between the two groups of vectors with 3 out of 4 applied models of compositional distributional semantics.

These results suggest that the models of compositional distributional semantics in combination with the measures for detecting semantic anomaly in AN combinations can be applied to the task we address in this work, therefore, we re-implement and extend the ideas presented in Vecchi et al. (2011). In particular, we use the 3 models of compositional distributional semantics – the *add*, *mult* and *alm* models. We do not use the *lm* model since it is based on an idea similar to the *alm* model, but performs worse on the anomaly detection task. We also follow the semantic space experimental setting and the evaluation procedure of Vecchi et al. (2011), but extend the set of measures of semantic anomaly. We discuss the application of these models to learner data in Chapter 5.

2.5 EDC systems evaluation

EDC systems are evaluated with the set of measures widely used in other NLP tasks – *accuracy*, *precision*, *recall* and *F-measure*. The key concepts for evaluating performance are the number of *hits*, *misses* and *false flags* that the system makes.

For ED, *hits* or *true positives* (*TP*) on the class of errors are the correctly identified errors, *misses* or *false negatives* (*FN*) are the errors not identified by the system, and *false flags* or *false positives* (*FP*) are the correct instances incorrectly flagged as errors by the system. *True negatives* (*TN*) – the correct instances classified as correct by the system – are relevant for estimating *accuracy* but not other measures. The number of *true/false positives* and *negatives* is estimated by comparing the difference between the original and the gold standard annotation with the difference between the original and the system's suggestion: if the gold standard is different from the original text, then there is an error in the original which is expected to be identified by the ED system. Table 2.8 illustrates this with an example. For detection, the exact match between the gold standard correction and the system's suggestion is not required (see the *TP* cell in Table 2.8): as soon as both differ from the original word, an error is flagged in the right place and is counted as a *hit*.

For correction, it is important that not only the error is identified, but that the correction also coincides with the gold standard one. Therefore, if an error is identified on the word

Category	Positive (P)	Negative (N)
True (T)	*Original:* I have tried a *classic* dance already. *GS:* I have tried a *classical* dance already. *System:* I have tried a *typical* dance already. => **a hit**	*Original:* They performed a *classic* Ceilidh dance. *GS:* They performed a *classic* Ceilidh dance. *System:* They performed a *classic* Ceilidh dance.
False (F)	*Original:* They performed a *classic* Ceilidh dance. *GS:* They performed a *classic* Ceilidh dance. *System:* They performed a *classical* Ceilidh dance. => **a false flag**	*Original:* I have tried a *classic* dance already. *GS:* I have tried a *classical* dance already. *System:* I have tried a *classic* dance already. => **a miss**

Table 2.8: *True/false positives* and *negatives* for error detection.

Category	Positive (P)	Negative (N)
True (T)	*Original:* I have tried a *classic* dance already. *GS:* I have tried a *classical* dance already. *System:* I have tried a *classical* dance already. => **a hit**	*Original:* They performed a *classic* Ceilidh dance. *GS:* They performed a *classic* Ceilidh dance. *System:* They performed a *classic* Ceilidh dance.
False (F)	*Original:* They performed a *classic* Ceilidh dance. *GS:* They performed a *classic* Ceilidh dance. *System:* They performed a *classical* Ceilidh dance. => **a false flag**	*Original:* I have tried a *classic* dance already. *GS:* I have tried a *classical* dance already. *System:* I have tried a *typical* dance already. => **a miss**

Table 2.9: *True/false positives* and *negatives* for error correction.

classic and the correction suggested by the system is *typical* rather than *classical* as in the gold standard, this is counted as a *miss* – see Table 2.9.

In this work, we focus on ED specifically. System performance evaluation is more difficult when it is subject to matching the correction in the gold standard, and it has been shown (Tetreault and Chodorow, 2008a) that some contexts license for multiple possible corrections and system performance is underestimated if exact match to only one correction is required. We believe that performance should be estimated on detection and correction separately. In addition, it has been shown (Leacock et al., 2009) that learners can benefit from ED even if the corrections provided by the system are not good. We discuss the error correction step in Chapter 7.

Learner data is typically highly skewed, with the number of errors even for the most frequent error types being relatively low: the error rate is usually about 5% only. This makes ED quite a challenging task, since the baseline – detecting no errors – is high from the beginning. For some error types, it is also hard to clearly define *TN*s which are important for the estimation of accuracy of an ED algorithm: for example, for determiners the *TN*s are the determiners correctly used by the learners and not detected by the system as errors. However, estimation of the number of correctly used determiners proves problematic as it is hard to determine the number of correctly used ∅ (*null*) articles: it is not clear whether each whitespace between the words, or every position before a noun or a noun phrase should be counted, and there is not much consensus on this issue (see Leacock et al. (2014, p. 34) for discussion). With respect to content word errors, this problem does not arise, since the number of *TN*s is simply the number of correctly used combinations of a certain type not tagged as errors by the ED system.

Using the number of the *hits*, *misses* and *false flags* returned by the system, the *accuracy* (*Acc*), *precision* (P) and *recall* (R) are estimated as:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.10}$$

$$P = \frac{TP}{TP + FP} = \frac{\#hits}{\#hits + \#falseflags} \tag{2.11}$$

$$R = \frac{TP}{TP + FN} = \frac{\#hits}{\#hits + \#misses} \tag{2.12}$$

Accuracy shows the proportion of items for which their status has been identified by the system correctly: it is the proportion of errors tagged as errors and correct uses tagged as correct. Precision shows how reliable the system is at detecting errors: how often what is identified by the system as an error is an actual error versus a false flag, while recall shows how many errors the system identifies among those present in the text. Precision determines reliability of the system, and there is clear evidence that precision is more important for learners and their progress than recall, so that most EDC systems aim for high precision (Chodorow et al., 2010; Nagata and Nakatani, 2010; Ng et al., 2014).

The F_1-*measure* is the harmonic mean of precision and recall, and both contribute equally to the equation. However, since more emphasis has been put on precision of EDC systems lately, $F_{0.5}$ has been used in the recent shared task (Ng et al., 2014) which puts twice the weight on precision than on recall:

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{2.13}$$

$$F_{0.5} = (1 + 0.5^2) \cdot \frac{P \cdot R}{0.5^2 \cdot P + R} \tag{2.14}$$

Some issues related to evaluation of EDC systems have been raised before. Chodorow et al. (2012) point out that EDC is more complicated than many NLP tasks: instead of a two-way correspondence between a system's output and a gold standard we are dealing with a three-way correspondence between an original writer's sentence, the annotator's correction which is taken to represent the gold standard and the system's output. It is not clear whether the gold standard based on annotator's correction is exhaustive, or whether some of the system's output not coinciding with the gold standard can be used to extend and amend it. We believe, that with respect to ED alone, the mismatch between the system's and the annotator's correction is not directly relevant as both would be considered as a hit as soon as they differ from the original (see §2.2).

Another issue that makes EDC a particularly challenging task is the high skew in the actual learner data: the goal of the EDC systems is to detect and correct errors, while errors as compared to non-errors have much lower frequency. The lower bound, or the majority baseline, is very high and unless an EDC system can reliably identify errors, it is practically useless, as any system that makes no corrections at all would outperform an EDC system that makes some wrong predictions. In particular, it is important that an EDC system has high precision on the class of errors, as it can be argued that *FP*s on the error class are more "costly" than *FP*s on the correct class (Nagata and Nakatani, 2010). *FP*s on the correct class (non-errors) represent errors missed by the system and tagged as correct, while *FP*s on the incorrect class (class of errors) represent some originally correct uses that are tagged by the system as errors. It can be argued that errors have higher informativeness: language learning quite often is based on errors and their correction, and it is important not to mislead the learners by drawing their attention to originally correct expressions as they might memorise them as being incorrect. In addition, if an originally correct expression is tagged by the system as an error, the learner is forced to look for a correction for an originally correct expression.

Helfrich and Music (2000) conducted a customer study to determine which error types grammar checkers should primarily focus on. Their study confirms that even native speakers, who presumably can distinguish between *TP*s and *FP*s returned by an EDC system, are less concerned about recall than they are about precision and the number of false flags. Users will often turn a grammar checker off due to a "spectacular false flag and/or annoyance", and errors on common words can make a bad impression of the system quality. Helfrich and Music (2000) concluded that a good balance should be found between reduction of the number of false flags and spotting and correcting the errors people make, however, increasing precision and decreasing the false flag rate should have higher priority than recall. In addition to annoyance by false flags, non-native speakers are also not always able to distinguish between correctly identified errors and false flags.

Nagata and Nakatani (2010) studied how EDC systems should be evaluated so that they maximise learning rate. As they noted, perfect EDC might not be attainable, but it could be assumed that learners still benefit from imperfect EDC. As a matter of fact, imperfect EDC can maximise learning rate as it can promote learners' thinking more than perfect EDC when all errors are detected and corrected so that a learner has nothing to do but just accept all the suggestions from the system. They explored whether a system that is recall-oriented and identifies more errors with some of them being originally correct instances, or a precision-oriented system which identifies fewer errors but those identified are true errors, maximises learning rate. They also questioned whether the F_1-*measure* is indeed the best evaluation measure for EDC systems.

The motivation for focusing on higher precision is that when only a limited number of errors is detected with high precision, learners can detect the other incorrect instances by examining the system's feedback and generalising it to more instances in their text, using their knowledge of English and consulting grammar books. These activities, it can be argued, lead to improvement in one's knowledge of English. In contrast, if a system detects many errors but with limited precision, learners focus on judging whether the given results are trustworthy or not, and would not learn much from such feedback.

Nagata and Nakatani (2010) focused on errors in articles and noun number. Learning activities consisted of essay writing, error detection and rewriting, and the learning effect was measured by the decrease in error rate. Four feedback conditions were compared: no-feedback, recall-oriented, precision-oriented, and human feedback taken to represent perfect ED. Recall- vs precision-oriented conditions correspond to the number of errors detected by the system. It was shown that the best condition was human-generated feedback, but the decrease in the error rate for the precision-oriented system was close to that for human feedback. The group with recall-oriented feedback showed even worse results than the group without feedback. This proves that imprecise feedback misleads learners more than no feedback. In the experiments of Nagata and Nakatani (2010), the precision-oriented condition corresponded to 0.72 system precision and 0.25 recall.

Several conclusions can be drawn from this study. First of all, precision-oriented feedback has a similar learning effect to the ‘perfect’ ED provided by a human annotator. In the situation where full and accurate ED by a human annotator is not available, an imperfect ED generated by a computer can be used as a substitute if it has high precision. In contrast, ED with high recall at the cost of lower precision is potentially harmful. As a consequence of this, systems should be both designed to favour precision and evaluated primarily with respect to their precision. For example, the recent shared task on EDC used $F_{0.5}$ favouring precision over recall (Ng et al., 2014).

At the same time, the results of this study should be accepted with caution, given that only a limited number of error types over a limited time period were investigated. We support the idea that an ED system should be oriented to high precision, but we recognise that more studies might be needed to investigate the learning effect on content word errors.

Chapter 3

Data collection and annotation

In this chapter we describe the datasets of correct and incorrect AN and VO combinations used in our experiments. In the CLC-FCE, errors involving incorrect choice of verbs, nouns and adjectives account for 11.75% of all errors. Among the different types of content word combinations with words of these parts of speech ANs and VOs are some of the most frequent ones.

We have adopted two different approaches to extracting the data, and as a result, we run our experiments on two different datasets. We use the CLC to extract all the examples of the AN and VO combinations. We have discussed the CLC in §2.1.1 and noted, that word combinations of a particular type can be extracted from the CLC using the error annotation scheme. The relevant errors for our task are coded with `R*`, where `R` denotes the error type *replace word*, while the second letter stands for the part of speech – `J` for an adjective, `N` for a noun and `V` for a verb. Examples of the incorrect combinations extracted from the data are given in Table 3.1.

The error-annotated part of the CLC can be used to directly extract erroneous examples with their corrections. The first dataset is extracted from the publicly-available CLC-FCE dataset (Yannakoudakis et al., 2011) (see §2.1.2). As we rely on the original error annotation for data extraction, we refer to this dataset as *uncontrolled* in contrast to our second dataset, where the examples have been selected in a more controlled way: for instance, the examples from the *uncontrolled* dataset contain some combinations previously seen in native English corpora such as the BNC (Burnard, 2007). The AN subset contains $4,681$ correct and 530 incorrect combinations, with $3,294$ of the correct ones and 286 of the incorrect ones attested in the BNC.[1] The VO subset contains $4,911$ correct and 789 incorrect combinations, with $3,997$ of the correct ones and 560 of the incorrect ones attested in the BNC. We discuss this dataset in more detail in §3.1.

We refer to the second dataset of AN and VO combinations as *controlled*: we extracted the word combinations from the full unannotated CLC, selecting only the combinations unseen in the BNC. Since we use unannotated learner data for this dataset, we cannot rely on error annotation but absence of a word combination in the corpus of native English is

[1] This dataset is available at `http://ilexir.co.uk/applications/adjective-noun-dataset/`.

Error code	Explanation	Example (error | correction)
RJ	incorrect adjective	**big variety | great variety*
RN	incorrect noun	*big *roll | big role*
RJ+RN	incorrect adjective and noun	**big *edge | broad brim*
RV	incorrect verb	**make photo | take photo*
RN	incorrect noun	*make *search | make research*
RV+RN	incorrect verb and noun	**make *store | develop story*

Table 3.1: Examples of the extracted incorrect word combinations.

a cue that the combination may contain an error. On the one hand, checking the word combinations against the BNC is used as a heuristic approach to collect learner errors from unannotated data. On the other hand, this dataset reflects an important property of human language: since language is productive, no corpus can effectively sample all possible content word combinations (Chomsky, 1957, p. 15). This problem arises in ED in learner data, since learners are creative and use a high number of word combinations that have never been seen before. From the educational point of view, it is desirable that an ED system does not "punish" language learners for their creativity: it should recognise incorrect word combinations but not flag acceptable ones as incorrect simply because those are unseen in native English data. This makes this dataset particularly challenging for ED algorithms, since algorithms that rely on purely statistical measures based on the corpus occurrence counts would not be applicable, and a more effective approach to ED in this data requires a semantic component to distinguish between correct and incorrect content word combinations.

We describe the *controlled* dataset in §3.2. §3.2.1 presents the general motivation for data collection. We have devised an annotation scheme for this dataset to distinguish between correct and incorrect combinations, where we also distinguish between their use in isolation and in their original contexts in the CLC. The annotation scheme is presented in §3.2.3. We have used a number of error codes to describe the type and probable reason for the error as well as its locus. We have also provided the most probable corrections for the incorrect word combinations. The data has been primarily annotated by a professional linguist, but to ensure that the devised annotation scheme is comprehensive and clear, a small subset of the *controlled* dataset (about 100 examples for both AN and VO combinations) has been first annotated by three annotators. §3.2.4 discusses the results of this annotation step, specifying inter-annotator agreement and statistics with respect to different error types. The *controlled* dataset contains 798 AN and 800 VO combinations, with error annotation and corrections provided.

3.1 CLC-FCE dataset

We have used the error annotation provided with the CLC-FCE (Yannakoudakis et al., 2011) to divide the extracted word combinations into the subsets of correctly used ANs and VOs and those that contain errors in the choice of component words.

For certain word combinations, all occurrences are annotated as errors. Example (3) illustrates a frequent error committed by language learners using an AN **big variety* where *big* is used instead of more appropriate adjectives like *wide* or *great*, while example (4) illustrates another frequent error in a VO **answer information* where *answer* is too specific and literal for the action described.[2]

(3) The group asked me to make sure that in the fridge they always could find a <NS type="RJ"><i>*big*</i><c>*wide*</c></NS> *variety* of drinks including honey and lemon for the throat.

(4) I am writing to you because I would like to <NS type="RV"><i>*answer*</i> <c>*provide*</c></NS> the *information* that you ask for.

Such combinations in our dataset are unambiguously annotated as errors. However, certain word combinations are used correctly in some contexts and incorrectly in others, and their annotation is more debatable. For example, *best date* is appropriate in contexts where it is used to point to a particular date. Example (5) provides a context in which the use of the *best date* is acceptable:

(5) Firstly, the *best date* for me to travel will be in the second week of July because I will be very busy at any other <NS type="RN"><i>*date*</i><c>*time*</c></NS> for work reasons.

The second use of the noun *date* in the same sentence suggests that learners may also overuse *date* to denote *time* in general. Learners quite often misuse time-related nouns: for example, *moment*, *period*, *term* and similar nouns are frequently used incorrectly in cases where the general word *time* is more appropriate. Example (6) illustrates another case of an incorrect use of *date* instead of *time* within *best date*:

(6) Because of this I believe this is the *best* <NS type="RN"><i>*date*</i><c>*time* </c></NS> for me to travel.

Similarly, a VO combination *buy suit* can be used correctly in certain contexts to describe a specific situation of purchasing a suit, as illustrated by example (7):

(7) On the other hand it is a great achievement for you when you *buy* the *suit* that is perfect for everyone.

However, *suit* might be too specific a term to be used to denote buying clothes in general, and in such contexts it should be annotated as an error. Example (8) illustrates an incorrect use of *suit* when *outfit* is more appropriate:

(8) For example, imagine that you are a bride and you have to *buy* an appropriate <NS type="RN"><i>*suit*</i><c>*outfit*</c></NS>.

[2] These examples have been extracted from the CLC-FCE. Other errors except for the ones discussed are corrected using the provided error correction.

The question then arises how such combinations should be represented in the gold standard. One can adopt a *token-* or a *type-based* approach to annotation. Within a token-based approach, each occurrence of a word combination in each particular context of use can be treated independently of the others so that some of the instances would be annotated as correct and others as incorrect; in contrast, within a type-based approach a combination should be assigned to just one class.[3] The token-based approach would allow us to keep the *in-context* annotation: for example, we would distinguish the occurrences of *buy suit* in contexts (7) and (8) as a correct and an incorrect instance respectively. Such annotation is particularly useful for context-specific approaches to ED.

However, we experiment with more general approaches to ED that do not make use of context, and for such approaches it is more useful to assign each combination to only one class as per the type-based approach. One option is to put all combinations that are correct in at least some contexts in the 'correct' category to avoid possible overcorrection that might mislead the learner: thus, both *best date* and *buy suit* exemplified in (5) to (8) would be annotated as correct. Another option is to make use of the available learner data and make the gold standard represent the most frequent annotation: in this case, since both *best date* and *buy suit* are more frequently used incorrectly than correctly, both combinations would be included in the gold standard as errors.

The error annotation we rely on for this dataset is rather generic: it distinguishes between correct and incorrect word combinations and provides information on the locus of an error. In contrast, the annotation scheme that we apply to the *controlled* dataset (see §3.2.3) is also aimed at describing the most probable reason for an error.

Another important feature of this dataset is that it contains a wide range of word combinations, both attested and unattested or rare in the BNC. The question of why certain word combinations annotated as errors in the CLC are attested in the BNC deserves more detailed explanation. One group of such examples contains word combinations like *best date* or *buy suit* which can in appropriate contexts be correct and used by native English speakers, but are frequently misused by language learners as in examples 6 and 8 above.

Then, some word combinations are generally less frequently used even by native speakers, and less expected to be used by language learners as well, than their corrections. A typical example of this phenomenon in our data is a frequent confusion between *classic* and *classical*: while both adjectives within these confusion pairs can combine with certain nouns to produce phrases with a different meaning, for some of these pairs one combination might be clearly more widely used while another might have a very specific meaning. For example, both *classic music* and *classical music* are attested in the BNC. It is possible to find contexts in which the former is used appropriately, but such contexts all refer to a very specific use of the combination *classic music* – for example, *classic music for an event.* In contrast, *classical music* is a much stronger collocation and is more frequently used both in native and learner corpora: for comparison, *classic music* is seen 1 time in the BNC alone and 150 times in the BNC and ukWaC together, while *classical music* – 121 times in the BNC alone and $4,006$ times in the combined corpus. Even though *classic music*

[3] We treat error detection as a binary task as is a usual practice in the field in general. In spite of the fact that some work in the field advocate the use of a scale for error annotation and detection, the usefulness of such approach for language learning has not been proven (also see the discussion in §2.2).

is corpus-attested in native language, it is much less frequently used by native speakers, and most often it is used incorrectly by learners instead of *classical music*.

Finally, we also note that some of the occurrences of the AN and VO combinations attested in the BNC can themselves be quite rare word combinations which could cover figurative senses, uses as part of proper names,[4] or similar. We might then expect to see many infrequent combinations among those that are both corpus-attested and annotated as errors in the CLC. Indeed, such combinations from our dataset are, on average, less frequent in the BNC than those which are both correct and corpus-attested.

As one of our approaches to ED is based on compositional distributional semantics, it is important to outline certain differences between the datasets used in this research on ED and the research on semantic anomaly detection by Vecchi et al. (2011): Vecchi *et al.* have used a limited set of constituent adjectives and nouns and an approximately equal number of semantically acceptable and deviant combinations, whereas our dataset is more skewed towards correct combinations and covers a wider range of constituent words. In addition, some of the correct and incorrect combinations from our dataset are attested in the native English corpora. We take these factors into account, but we believe that our dataset reflects practical applications of semantic anomaly detection more closely. We describe our experiments using the semantically-based approach in Chapter 5.

We refer to this dataset as *uncontrolled* and use this term to denote that we do not control whether the combinations occur in the native English corpora.

3.1.1 CLC-FCE AN subset

The extracted AN subset contains $4,681$ correct and 530 incorrect combinations, where we take the type-based majority-based annotation for the gold standard: for example, the AN *best date* is annotated as an error because it is more frequently used incorrectly than correctly (see §3.1). $3,294$ of the correct and 286 of the incorrect ANs occur in the BNC.

The ANs in this subset are formed with $1,061$ distinct constituent adjectives and $1,335$ nouns. 6 adjectives are used in 100 to 200 ANs, with *good* (*good day* versus **good nature* | *good environment*), *new* (*new phone* versus **new manners* | *new ways*) and *big* (*big problem* versus **big variety* | *wide/great variety*) being the most frequent and having the highest number of examples. All other adjectives are used in 1 to 100 ANs, with 958 adjectives being used in 10 or less ANs, and 553 adjectives being exemplified by one AN each.

The 530 incorrect ANs contain combinations with 259 adjectives and 324 nouns. Adjectives *big* and *good* have the highest number of incorrect examples.

3.1.2 CLC-FCE VO subset

The VO subset contains $4,911$ correct and 789 incorrect combinations, with $3,997$ of the correct ones and 560 of the incorrect ones attested in the BNC. We have adopted the same type-based majority-based approach to derive the gold standard annotation.

[4] For example, a company name is not necessarily a proper English phrase. We have encountered an AN *Funny Pub* in the learner data used as a name of a pub rather than in its literal meaning.

The VO subset contains a comparable number of combinations to the number of AN combinations, but is less diverse in terms of constituent verbs: it contains combinations with only 603 distinct verbs and 1, 586 nouns. The verb *have* (*have break* versus **have conclusion* | *come (to) conclusion*) is used in the highest number of VOs (521) in this subset. The other 7 verbs with the highest number of examples – in the range of 100 to 200 instances – include *make* (*make claim* versus **make competition* | *enter competition*), *see* (*see bird* versus **see pleasure* | *take pleasure*), and *take* (*take advice* versus **take breakfast* | *have breakfast*). For 490 verbs the number of examples ranges from 1 to 10, while for 235 verbs there is only one example.

The 789 incorrect VOs contain combinations with 241 verbs and 492 nouns. The verbs *have* and *make* have the highest number of incorrect examples.

3.2 Annotated dataset

We collected our second dataset in a more *controlled* way, and in this section we review the underlying principles of our approach to data collection. We have also devised an annotation scheme aimed at not only identifying each word combination as either correct or incorrect, but also suggesting the most probable reason for the error committed. In this dataset release, we include error-coded examples of word combinations in their original contexts of use in the CLC, supplemented by corrections and metadata information.[5]

3.2.1 Overview of the dataset

As opposed to the *uncontrolled* dataset presented in §3.1, we use a different approach to extracting the examples for this dataset. We have extracted the ANs and VOs from the full unannotated CLC and focused on the combinations that do not occur in a corpus of native English (in this research, we use the BNC). We refer to this dataset as *controlled* and the factor that we control for is the non-occurrence of the content word combinations in the native English corpus. The underlying principles behind this data collection are:

1. **The nature of learner errors**: Most of the state-of-the-art approaches (see §2.3.2) to EDC in content word combinations rely on comparison of occurrence frequency of the original (and, possibly, incorrect) combination and its alternatives. For example, adjectives *strong* and *powerful* are close in meaning and might be confused with each other by language learners. As such, they belong to a confusion set [*strong, powerful*]. We might see an AN **powerful tea* in the learner data, and using such frequency-based approaches run on the set of alternatives [*strong tea, powerful tea*] detect and correct the error. In that case, “correctness” of the word combination is equated with the fluency based on the frequency of occurrence in native data. Although this heuristic may be effective in detecting less fluent word combinations, a question arises whether an error in the use of content words should be defined using fluency. Not every combination that is less fluent than some of its alternatives

[5] Currently, the dataset is available at `http://www.cl.cam.ac.uk/~ek358/data/`

is incorrect itself. For example, the adjectives *appropriate* and *proper* are similar in meaning and can often be used interchangeably. *Appropriate concern* is correct but has lower collocational strength than its alternative *proper concern*,[6] which means that in native English the latter is more fluent than the former. If we were to follow the fluency-based approach to ED, we would have to tag an originally correct *appropriate concern* as an "error". Similarly, we would have to tag *proper discount* as an "error" since its alternative *appropriate discount* is more fluent. At the same time, Chomsky's famous *colourless green ideas* example will be corpus-attested despite being semantically deviant. We maintain that, from the educational point of view, flagging an originally correct combination as an error is more harmful and misleading for language learners than missing a possibly incorrect one (see §2.5). We are primarily interested in approaches that are not based on pure corpus-based comparison. The dataset consisting of examples from learner data that are not attested in a native corpus of English presents a good test set, since correctness of such word combinations cannot be assessed using corpus statistics only.

2. **Creativity in learner language versus data sparsity**: Another property of natural language that is illustrated by this dataset is that the number of possible content word combinations is too big for any corpus to cover all of them (Chomsky, 1957, p. 15). However large the corpus of native language is, some correct and semantically acceptable content word combinations will not occur or will have low counts. The reliability of corpus statistics for a word combination drops as the length of the combination grows. This shows that algorithms that rely less on (co)occurrence statistics and more on deeper language processing are needed. This issue is relevant to learner data and EDC as learners are creative in their use of language and they may use word combinations that are not attested in a corpus of native English. Some of those combinations would be incorrect due to learners' lack of understanding how to choose words correctly as, for example, **big knowledge* | *extensive knowledge*, **deep regards* | *kind regards*, or **catch job* | *get job*, **join seminar* | *attend seminar*. Others would be acceptable but not covered by the corpus. Many of the word combinations used by native speakers would also not be covered by the corpora of native English, but it is more usual to attribute those to creative use of language rather than ignorance (see §1.1). Our goal in this research is an ED system that does not "punish" language learners for their creativity, and is able to make semantically-motivated decisions.

3. **The challenge for ED and EDC algorithms**: The dataset consisting of only previously unseen word combinations is more challenging for ED and EDC systems, as such systems should involve a semantic component and deeper language analysis.

4. **A new test bed for compositional distributional semantic models**: While ED in content word combinations in learner language is the focus of our research, we think that this dataset can also be used as a new test bed for compositional distributional semantic models, since it consists of examples extracted from actual learner data (that is, natural environment) as opposed to artificial examples that are often used to test these models (Vecchi et al., 2011; Mitchell and Lapata, 2010).

[6] Collocational strength has been estimated using Normalised Pointwise Mutual Information.

We have designed this dataset to illustrate the most typical confusions that occur in learner language and have focused on the most frequently misused adjectives and verbs (see §3.2.2). The examples are extracted from the CLC, with the information about the examinations and the learners, including their L1s and level of language proficiency, retained.

3.2.2 Data extraction and preprocessing

In collecting this dataset we had two major goals:

(i) to collect a set of previously unseen AN and VO combinations;

(ii) to include the combinations that illustrate typical learner errors.

To ensure that the collected data complies with goal (i), we have used the BNC. To ensure that it complies with goal (ii), we first analysed the errors committed by language learners in the publicly available CLC-FCE dataset. We have focused on the most frequently misused adjectives and verbs.

For the ANs, we have focused on the combinations where adjectives are used inappropriately and extracted examples tagged with the `RJ` error code. Then, we filtered out the adjectives with low error rates: if some combination more or equally often appears in contexts where it is considered to be correct, it was excluded from our set. Next, rare adjectives which appeared in less than 4 examples in total were filtered out. As a result, we obtained a set of 61 adjectives including *big* and *good* that are frequently misused by language learners (see §3.1.1), adjectives semantically close to them like *large*, *great* and *high*, as well as some other adjectives frequently confused with each other like *classic* and *classical*, to name just a few. Morphological forms like *good* and *best* are treated as two distinct adjectives rather than the forms of the same adjective, as they have different confusion patterns: while *good* might be overused by language learners instead of more specific adjectives, *best* is most frequently confused with *favourite* as in *my *best hobby*.

We have used the set of 61 adjectives to extract the AN examples from the unannotated part of the CLC, checking that they do not occur in the BNC and filtering out combinations where the noun is not attested in the corpus. We have also checked for the alternative spelling of the nouns, e.g. American versus British English spelling. The final set contains 798 corpus-unattested ANs.

We have applied the same approach to extracting VO combinations. For that, we have first identified a set of 77 verbs that are most frequently misused in the CLC-FCE dataset, including *have*, *make*, *do*, *say*, *tell*, *see*, *look*, to name just a few. We have extracted 800 corpus-unattested VOs from the unannotated part of the CLC.

3.2.3 Annotation scheme

We have devised an annotation scheme aimed at:

(i) classifying word combinations as correct or incorrect out of any particular context of use as well as in their original context of use extracted from the CLC;

(ii) describing, in case of an error, the most probable reason for the error: e.g., whether we could identify an alternative word close in meaning or in form to the originally used incorrect one that would be a suitable correction;

(iii) suggesting, in case of an error, the most probable correction.

Our observations of the most typical learner errors combined with our analysis of the *uncontrolled* dataset (see §3.1), which was extracted from the error-annotated CLC-FCE, helped us devise an annotation scheme that we consider appropriate and comprehensive for annotating errors in content word combinations. The annotation scheme is designed to be general enough so that it can be applied to any type of content word combinations with only minor changes. For example, we first completed the annotation of AN examples using this scheme, and then applied it to annotating VO examples.

The dataset was primarily annotated by a professional linguist, Diane Nicholls. To ensure that the annotation scheme is clear and efficient, the dataset was split into 100 and 698 combinations for the ANs and 100 and 700 combinations for the VOs, and in each case the 100 combinations were first annotated by four annotators including Diane Nicholls. Two of the four annotators (referred to as annotators 1 and 3 in the discussion below) are native speakers of English, while the other two speak English at the proficient level. We measured inter-annotator agreement (see §3.2.4), and discussed all controversial cases before the main annotator proceeded to annotating the full set. Annotation was performed using a tool developed for these purposes by Øistein Andersen.

Out-of-context versus in-context annotation

We distinguish between *out-of-context (OOC)* and *in-context (IC)* annotation. Previously, Lee et al. (2009) have used a similar approach presenting items without context and in context to study how annotators' decisions about article use and noun number change depending on the availability of surrounding context. We have briefly mentioned in §3.1 that certain word combinations might be correct in some contexts and incorrect in others, and that some combinations, even though being generally correct, might be overwhelmingly used incorrectly by language learners. We supported our claim with two examples – the AN *best date* and the VO *buy suit*.

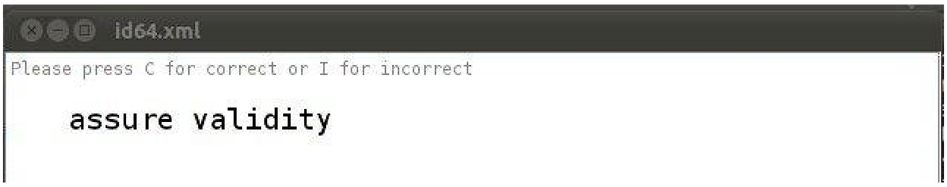
Combinations which may appear to be correct when considered out of their original context of use, because there might be other contexts where the same combination would be appropriate, are quite frequent in our data. For example, *classic dance* is correct out of context because one could imagine using it in a context like:

(9) They performed a *classic* Ceilidh *dance*.

However, in practice, the AN *classical dance* is used much more frequently both in learner and in native English data, and when learners use *classic dance* what they actually mean is *classical dance* as in, for example:

(10) I have tried a rock'n'roll dance and a **classic|classical dance* already.

We designed the annotation scheme to allow for both *OOC* and *IC* annotation. Thus, we created a two-level annotation scheme, and the annotators are first presented with a word combination and asked to tag it as correct or incorrect depending on whether they can think of some appropriate contexts of use for it. Figure 3.1 shows the annotation tool screen presented to the annotators at this stage.

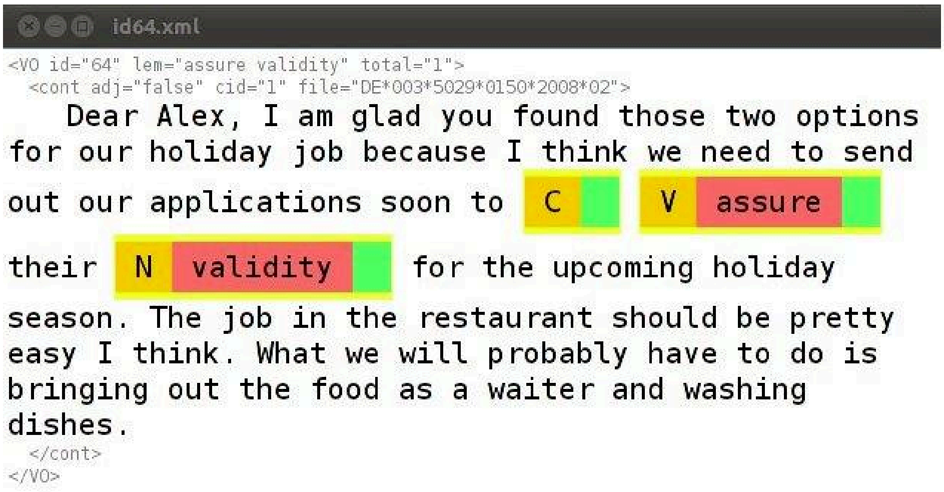


```
Please press C for correct or I for incorrect

    assure validity
```

Figure 3.1: The *out-of-context* annotation box.

At this step, a word combination is presented to the annotator in isolation and the annotator is prompted to assess whether it can be considered correct or incorrect in general. The relevant tags in the annotation scheme are `C` for *correct* and `I` for *incorrect*. The annotator is asked to press either `C` or `I` on the keyboard. Once the *OOC* annotation has been entered, the annotator is presented with the next screen – shown in Figure 3.2 – where the same combination is presented in its context of use from the CLC and the annotator is asked to annotate it with respect to this context.



```
<VO id="64" lem="assure validity" total="1">
  <cont adj="false" cid="1" file="DE*003*5029*0150*2008*02">
   Dear Alex, I am glad you found those two options
for our holiday job because I think we need to send
out our applications soon to  C    V assure
their  N validity   for the upcoming holiday
season. The job in the restaurant should be pretty
easy I think. What we will probably have to do is
bringing out the food as a waiter and washing
dishes.
  </cont>
</VO>
```

Figure 3.2: The *in-context* annotation box.

Tags `C` and `I` are meant to characterise the word combination in general, and at the second step when the annotator is presented with the original context of use for the combination they might realise that (a) they have not thought of this particular use and would like to change the tag `I` (generally incorrect) to `C` (generally correct); (b) the combination is inappropriate for this dataset, for example, due to misparsing; (c) the context is too ungrammatical or nonsensical making it impossible to understand the learner's communicative intent. In case (a), the annotator can return to the previous screen and change the annotation. Similarly, in cases (b) and (c) they can go back and add tag `O` for the combinations that should be excluded from the dataset, or `U` for the combinations in *ungrammatical* non-sensical contexts.

If the combination is assigned tag `C` when considered *out-of-context*, it might be correct both *out of* and *in context*. In that case, the annotator simply saves the annotation and proceeds to the next combination. However, if the combination is used inappropriately *in context*, and also if it has been assigned `I` *out-of-context*, the annotator is asked to assign it one of the error tags aimed at defining the *most probable* reason for the error committed and the *most probable* correction. Such annotation relies on the annotators' ability to understand the learner's communicative intent, and to ensure consistency and help the annotators come up with the *most probable* corrections we have provided them with annotation guidelines on how to treat some debatable cases.

In-context error tags

We note that learners rarely commit inexplicable errors. For example, if the AN combination that complies with the learner's communicative intent is A_1N_1 and the one that is actually used is A_2N_1, A_1N_2 or A_2N_2 (either adjective or noun or both are used incorrectly), there is a high chance that the adjective A_2 and/or noun N_2 are related to the adjective A_1 and/or noun N_1 that express the learner's communicative intent either in form or in meaning. Even if the direct link is not easy to establish, we hypothesise that such a link is present in the learners' mental lexicon, for example, through L1-transfer, via some non-classical semantic relations, and so on. We included in our annotation scheme a specific tag to denote that no *obvious* relation between the used word and the proposed correction can be detected by the annotator, but the annotators were also asked to provide their comments in case they can identify language transfer or non-classical semantic relation. For example, **good humor* has been annotated as an error with the correction *good mood*. The two nouns are not semantically related in English, but knowing that the learner's native language is Portuguese, one could reliably identify L1-transfer. Since we cannot expect our annotators to be able to identify L1-transfer in all the cases where it occurs, this is included in the annotation scheme in the form of optional comments.

Our annotation scheme for the word level *IC* annotation is based on the relations that can be established between the used words and the suggested corrections. We hypothesise that in most cases the learner's communicative intent can be reliably detected and the link between what is used and what is meant can be established. The annotators were asked to provide error codes in the yellow field after PoS tag `V`, `J` and/or `N` (see Figure 3.2). The corrections and comments should have been provided in the green field.

Our annotation scheme contains the following *IC* error tags:

- tag `F` is used when the confusion results from a similarity in the form of two words bearing a morphological or derivational relation to each other, or just being close in pronunciation or spelling.[7] For example, **classic* and *classical*, **adapt* and *adopt* are annotated with `F`.

[7] Dahlmeier and Ng (2011a) in their experiments distinguish between spelling and homophones whereas we combine the two cases. Our study showed that they are too similar to be reliably distinguished.

- tag S is used when the confusion results from similarity in meaning. We limit the set of possible relations to synonyms and hypo-/hypernyms, as other relations appear to be much rarer and much harder to reliably detect.[8] This tag is used for such cases as **big/large quantity* or **big/great importance* where learners are not able to distinguish between synonyms and near-synonyms and choose an appropriate one from the set, or **big/long history* and **large/broad knowledge* related to the fact that certain high-frequency adjectives like *big* and *large* encompass a variety of meanings also covered by more specific adjectives like *high*, *wide* or *broad.* Learners' lack of intuition about which of these more specific adjectives should be chosen results in them using the ones with more general meaning. For VOs, tag S is used for confusion pairs *acquire* and *get*, *achieve* and *reach*, *tell* and *say*, to name just a few.

- tag N is reserved for the cases where no obvious link can be established between the used word and the suggested correction. Annotators are encouraged to comment on such cases if they can identify a possible reason for the error. For example, *good *humor/mood* and **want/have wish* are annotated with N.

- tag Y is used on nouns and noun phrases which are related via uncolloquial metonymy: this is quite a wide-spread phenomenon, when due to the selectional preferences or subcategorisation frame of the verb, the noun used in the original needs to be expanded to a noun phrase. This might also be related to L1-transfer or false friends. Examples of such errors include *great *income/source of income*, *typical *interest/places of interest* or *solve *traffic/traffic problems.*

- tag M is added to the verb annotation if the subcategorisation frame for the verb should be changed, for example, when a direct object phrase should be changed to indirect object. This type of error results from learners' lack of knowledge about the correct subcategorisation frames of English verbs, and possibly also from interlingual differences. For example, tag M is used for **ask/ask for consideration.*

- tag D is added to the general (phrase) annotation in cases when the whole AN or VO should be substituted with a single noun or verb, respectively. These cases exemplify errors caused by verbosity in an attempt to describe something for which a learner has no suitable word in their mental lexicon (see §1.2), for example, **economical boost – economy*, **get achievement – achieve* and **make caution – warn.*

We tried to standardise the annotation procedure in order to reduce the possible disagreement between different annotators. We introduced a number of annotation rules which impose order on how the tags should be applied. Tags F, S and N may be used for both phrases tagged as correct or incorrect *OOC*, and on any or on both words in a word combination. When several corrections are possible and they involve different words, the annotators were instructed to choose a correction and error annotation for the adjective in ANs and verb in VOs, since we have selected the phrases with the most frequently misused adjectives and verbs in the first place.

[8] In the survey by Mohammad and Hirst (2007) these relations are also the ones considered most prominent, however, other non-classical relations which might connect two words are also mentioned.

When a combination can be corrected in several ways, we impose the following order on the tags: `F` > `M` > `Y/D` > `S` > `N`. Following our intuition that the used words and the ones that comply with the learner's communicative intent are most often related to each other, we suggest that the annotator should only use tag `N` and an unrelated correction when they cannot find any related ones. We prioritise structurally minimal corrections, therefore in the tag hierarchy `F`, `M`, `Y` and `D` precede `S`.

Form-related corrections are preferred over semantically related ones as we assume that in the learner's lexicon the form-related words are chosen more eagerly than the semantically related ones. This assumption is based on the observation that in order to confuse two semantically related words, a learner should know both words and their meanings, whereas in the case of form-related confusions a learner might know of the existence of two words but not necessarily know the meaning of both. We assume that the latter occurs more frequently than the former. This is also in accordance with the annotation guidelines adopted for other learner corpora, including the CLC and NUCLE. In the example given in §2.1.1 the preferred correction *He said to me that* for **He said me that* will be annotated with `M`, while the less preferred one *He told me that* with `S`.

3.2.4 Analysis

Next we discuss inter-annotator agreement on these tasks, along with relevant statistics such as the distribution of the correct and incorrect examples and the use of different error tags.

Inter-annotator agreement

We measure inter-annotator agreement using Cohen's κ (see §2.2), and report the observed agreement and κ values for each pair of annotators, as well as the average agreement and κ. These values calculated on the subset of 100 ANs and 100 VOs annotated by four annotators show how applicable the annotation scheme is. They also show the natural difficulty of the task, so we use the average observed agreement as the upper bound for the ED algorithm.

A final annotation for our 100 examples was arrived at by choosing the majority vote. Any ties were further discussed by the annotators until agreement was reached. We also measure and report the agreement between the final annotation on the 100 examples arrived at by majority vote and discussion, and the professional annotator, Diane Nicholls, who has further annotated the full set of AN and VO examples. High agreement and κ values would clearly show that the annotator finds the scheme clear and easy to apply.

Since we use binary classification for ED and contrast correct with errorful cases, we convert all the specific error-type annotations to the binary case and measure inter-annotator agreement for the OOC and IC cases on the binary scale. For the OOC case it amounts to comparing the `C` (correct) to `I` (error) annotations, while for the IC annotation only the cases annotated with `C-J-N` or `C-V-N` are counted as 'correct' and all the others represent the 'error' class. Tables 3.2 and 3.3 present the observed agreement/κ values for the OOC and IC annotation for the ANs, while Tables 3.4 and 3.5 present the results

Annotators	An$_2$	An$_3$	An$_4$
An$_1$	0.86/0.6392	0.85/0.5960	0.86/0.6171
An$_2$	–	0.89/0.7304	0.92/0.8020
An$_3$		–	0.81/0.5123

Avg agreement = $\mathbf{0.8650 \pm 0.0340}$
Avg κ = $\mathbf{0.6500 \pm 0.0930}$
=> **Substantial agreement**

Table 3.2: Observed agreement/κ values for AN combinations, OOC annotation.

Annotators	An$_2$	An$_3$	An$_4$
An$_1$	0.76/0.5200	0.74/0.4800	0.72/0.4400
An$_2$	–	0.76/0.5185	0.78/0.5565
An$_3$		–	0.72/0.4309

Avg agreement = $\mathbf{0.7467 \pm 0.0221}$
Avg κ = $\mathbf{0.4917 \pm 0.0463}$
=> **Moderate agreement**

Table 3.3: Observed agreement/κ values for AN combinations, IC annotation.

for the OOC and IC annotation for the VOs. The bottom part of the Tables reports the average values used as upper bounds in further experiments, and summarises the level of agreement achieved. We report the mean values ± standard deviation.

The observed agreement between annotators on the ANs is quite high, setting the upper bound for the OOC annotation at 0.8650 and for the IC annotation at 0.7467. At the same time, the κ values can be interpreted as showing substantial and moderate agreement on the two types of annotation, with all the κ values being statistically significant at $P < 0.001$ (z ranging from higher than 4 to higher than 6 on both OOC and IC annotation). The fact that the agreement is not perfect can be explained by the natural difficulty of the task. Most of the disagreement between annotators comes from making different judgements about the ANs rather than from misinterpreting the annotation scheme or misunderstanding the guidelines (also see the discussion on the results reported in Table 3.8). Therefore, we consider the scheme itself to be reliable and comprehensive.

The observed agreement between annotators on the VO combinations is also high, but additionally we note that it is significantly higher on IC annotation for VOs than on OOC VO and IC AN annotation. The upper bound for OOC annotation is set at 0.8217 and for IC annotation it is 0.8467. The κ values show substantial agreement on both types of annotation, with all the κ values being statistically significant at $P < 0.001$ (z ranging from higher than 5 to higher than 7 on both OOC and IC annotation).

We report the observed agreement and κ values for the agreement between the final annotation and Diane Nicholls in Table 3.6. The values show that she was in substantial to almost complete agreement with the final annotation for the 100 examples of both AN and VO combinations. This shows that she found the annotation scheme clear and comprehensive.

The agreement values obtained are on a par or higher than the agreement values reported for annotating other error types and learner corpora (for example, Dahlmeier et al. (2013)).

Annotators	An_2	An_3	An_4
An_1	0.83/0.6605	0.78/0.5586	0.82/0.6051
An_2	–	0.87/0.7402	0.83/0.6611
An_3		–	0.80/0.5974
Avg agreement = $\mathbf{0.8217 \pm 0.0279}$ Avg κ = $\mathbf{0.6372 \pm 0.0585}$ => **Substantial agreement**			

Table 3.4: Observed agreement/κ values for VO combinations, OOC annotation.

Annotators	An_2	An_3	An_4
An_1	0.81/0.6045	0.82/0.6264	0.80/0.5920
An_2	–	0.89/0.7627	0.89/0.7710
An_3		–	0.87/00.7294
Avg agreement = $\mathbf{0.8467 \pm 0.0377}$ Avg κ = $\mathbf{0.6810 \pm 0.0751}$ => **Substantial agreement**			

Table 3.5: Observed agreement/κ values for VO combinations, IC annotation.

Setting	OOC	IC
AN	0.93/0.7993 ($z = 5.8499$) => **Substantial agreement**	0.91/0.8200 ($z = 8.2000$) => **Almost perfect agreement**
VO	0.85/0.6966 ($z = 6.8884$) => **Substantial agreement**	0.84/0.6679 ($z = 6.4403$) => **Substantial agreement**

Table 3.6: Observed agreement/κ values for the professional annotator vs. final annotation.

Statistics on the AN and VO combinations

Table 3.7 presents the distribution of the combinations that are tagged as correct and incorrect in the data by the annotators, both OOC and IC. We report the figures on the smaller set of 100 examples annotated by 4 annotators for comparison.

The first observation that can be made is that the statistics on the smaller set of 100 AN examples is noticeably different from that on the full AN dataset. The smaller subset contains a higher proportion of incorrect examples, both OOC and IC: the proportion of correct AN combinations in the full dataset is at least 10% higher than that for the 100 examples. The statistics for the VO combinations is more consistent, with the proportion of correct and incorrect examples being very similar in both the smaller subset and the full VO dataset. As we applied the same approach to extracting the 100 examples for the smaller AN and VO subsets, we believe that these subsets give a fair representation of the examples in the full datasets. The annotation for the smaller subsets and the full datasets was arrived at using the same approach for ANs and for VOs. Therefore, we can assume that the difference in proportion is due to the difference in the nature of errors in the two types of combinations, with the errors in VOs being more systematic than those

Type	Subset	OOC	IC
ANs	100	67.44% correct 32.56% incorrect	39.53% correct 60.47% incorrect
	All	**78.89**% correct 21.11% incorrect	**50.84**% correct 49.16% incorrect
VOs	100	55.93% correct 44.07% incorrect	36.44% correct 63.56% incorrect
	All	**55.57**% correct 44.43% incorrect	39.14% correct **60.86**% incorrect

Table 3.7: Distribution of correct versus incorrect instances in the data.

in ANs. We can also assume that, even though we used the same sampling procedure to extract the 100 examples for ANs and VOs, it might still be the case that the AN subset contains different proportion of incorrect combinations than the full AN dataset.

We have noted before that the difference in annotation might also be caused by different beliefs of the annotators about what is acceptable in English. In Table 3.8 below we report the percentage of the AN and VO combinations in the subsets containing 100 examples annotated as incorrect by each annotator originally (i.e., before any further discussion). We see that, on the average, native speakers (annotators 1 and 3) were more permissive than non-native speakers (annotators 2 and 4), as they annotated less combinations as incorrect. We note that since the full datasets were annotated by a single annotator (annotator 1), the difference between the annotation on the subset of 100 AN examples and the full AN dataset discussed above might also be caused by a particular approach adopted by this annotator towards annotating learner errors. Comparison of Table 3.7 and Table 3.8 shows that the proportion of the ANs annotated as incorrect in the full dataset (21.11% OOC and 49.16% IC) and the proportion of the ANs annotated as incorrect by annotator 1 (22% OOC and 50% IC) are very close. At the same time, the proportion according to the final annotation of the 100 ANs (32.56% OOC and 60.47% IC) is closer to the average number of ANs annotated as incorrect by the other 3 annotators (27.67% OOC and 55.33% IC). This can explain the difference observed in the annotation of the subset of 100 ANs and the full AN dataset. However, we also note that the annotation of VO examples performed under similar conditions does not follow this pattern.

The differences observed on the AN datasets as opposed to the relative stability of the annotation for the VO combinations suggest that acceptability of AN combinations is more controversial than acceptability of VO combinations. Given the difference in the distribution of the AN combinations annotated as incorrect in the smaller subset and in the full dataset, the question that needs to be answered is whether this is problematic for the experiments we perform on this data. The observed differences seem to suggest that the main annotator might have been more permissive towards potential learner errors than other annotators, and if we used more annotators on the full dataset, more ANs might have been annotated as incorrect. However, we do not consider the lower number of ANs annotated as incorrect problematic for our experiments, since as we discuss in §1.2.2, our goal is to avoid overcorrection and to develop a system aimed at high precision. Given that the professional annotator considers a higher number of ANs in learner writing acceptable, we believe that those ANs should not be classified as errors.

Annotators	ANs		VOs	
	OOC	IC	OOC	IC
An$_1$	22%	50%	46%	57%
An$_2$	30%	52%	51%	64%
An$_3$	27%	54%	42%	57%
An$_4$	26%	60%	48%	63%
Avg (1-4)	26.25 ± 2.85%	54.00 ± 3.74%	46.75 ± 3.27%	60.25 ± 3.27%
Avg (2-4)	27.67 ± 1.70%	55.33 ± 3.40%	47.00 ± 3.74%	61.33 ± 3.09%

Table 3.8: Proportion of combinations judged to be incorrect by the annotators.

In general, a higher number of VO combinations, both OOC and IC, are incorrect compared to AN combinations. The majority baselines for further experiments are set as the distribution of the most frequent class in the full dataset: for the AN combinations it is set to 78.89% (correct ANs) OOC, and 50.84% (correct ANs) IC, while for the VO combinations it is set to 55.57% (correct VOs) OOC, and 60.86% (incorrect VOs) IC. These are marked in bold in Table 3.7.

It is also interesting to note that quite a high number of ANs are judged to be correct OOC both on the 100 examples by 4 annotators (67.44%) and on the rest of the dataset by the single annotator (78.89%). When the same combinations are presented in their contexts of use, about 40% of those judged to be correct OOC are annotated as incorrect in context: a drop from 67.44% correct OOC to 39.53% correct IC shows that 41.38% of the generally correct ANs are incorrectly used in context, and the drop from 78.89% correct OOC to 50.84% correct IC on the full dataset corresponds to 35.56%.

The VO dataset is different in that respect: the proportion of VO combinations judged to be incorrect even out of their context of use is high – 44.07% and 44.43%. When the combinations that are judged to be correct OOC are presented to the annotators in their contexts of use, the drop in the number of correct combinations in the 100 examples subset and the full dataset corresponds to 34.85% and 29.57% respectively.

Tables 3.9 and 3.10 show the distribution of error tags with respect to the words used incorrectly (an adjective/verb, a noun, or both), and the reason for the confusion. We again note that the statistics are more consistent on the VOs than on the ANs. Most errors in the AN and VO combinations result from the incorrect choice of the adjective or the verb, and semantically motivated confusions are responsible for the majority of errors in ANs. This corresponds to the results reported in Ramos et al. (2010) and Liu (2002). For the VO dataset, however, the number of errors resulting from selecting a (seemingly) unrelated word is higher than that for the errors resulting from choosing an incorrect but semantically related word: 37.64% as opposed to 25.87% of cases.

Metadata statistics

We also collected statistics on the distribution of L1s in the data as well as the examination types and examination years for the essays used to extract the ANs and VOs for the datasets (see Tables A.1, A.2 and A.3 in Appendix A).

Type	Component	100	All
ANs	adjective	76.92%	66.50%
	noun	18.46%	27.66%
	both	4.62%	5.84%
VOs	verb	79.37%	76.28%
	noun	19.05%	19.37%
	both	1.58%	4.35%

Table 3.9: Distribution of errors on the components within combinations.

Type	Error	100	All
ANs	S	51.52%	56.20%
	F	43.94%	27.85%
	N	4.54%	15.95%
VOs	S	31.25%	25.87%
	F	20.31%	22.59%
	M	20.31%	13.90%
	N	28.13%	37.64%

Table 3.10: Distribution of error types in the data.

We show that our examples represent a wide range of L1s. Since we have not controlled for the L1s, some L1s are more widely represented in our dataset reflecting the natural distribution in the unseen combinations. This information can be used in further experiments on L1-related errors.

We also show that the essays used to extract the examples cover a wide range of examinations at different *CEFR* (*Common European Framework of Reference for Languages*) levels – from basic to proficient users.[9] We note that many AN and VO combinations unattested in the native corpora come from essays written by learners assumed to be at higher levels of language proficiency (see the high percentage of essays at the C1 and C2 levels in Table A.2). This can be explained by the fact that advanced learners are more creative in their use of language and construct more content word combinations than learners at lower levels. Learners at higher levels still produce a substantial number of errors: for example, Leacock et al. (2010) note that collocations are difficult for language learners at all proficiency levels including even advanced learners.

Table A.3 shows that the essays used for compiling the dataset cover examination years from 1993 to 2009.

[9] See `http://www.cambridgeenglishteacher.org/what_is_this` for more information on the CEFR levels, and `https://www.teachers.cambridgeesol.org/ts/exams` for the information on Cambridge English examinations and their correspondence to the levels.

Chapter 4

Baseline system

In §4.1, we discuss the theoretical background and motivation for using a system that is based on previous research in ED for content word combinations as a baseline (see §2.3.2). The practical implementation issues are presented in §4.2. We apply this system to both CLC-FCE and the annotated datasets, and the results are presented in §4.3. We show that the system fails to distinguish between correct and incorrect combinations in the controlled annotated dataset which contains previously unseen examples only. We analyse the results and draw conclusions about this system's performance in §4.4.

4.1 Theoretical background

In §2.3.2, we reviewed the previous approaches to EDC in content word combinations, and mentioned that most of them follow a three-step algorithm. We implement a simple system that is based on a similar approach, and use the results as a baseline in this project. The system has clear motivation and can be used as a reasonable baseline for any further ED approaches based on more sophisticated methods. However, it is important to show when this simple approach works well and when it cannot properly address the problem. We maintain that a system based on this approach will have the following limitations:

- It is common practice to use a reference database of known or previously seen miscollocations (Shei and Pain, 2000; Chang et al., 2008), but an EDC system relying on database lookup can only deal with a finite set of errors.

- Most approaches reduce the set of possible alternatives to semantically related words or to synonyms only, while other possible reasons for confusion are not considered. Most often, the synonyms are extracted from manually-created resources such as WordNet or thesauri (Shei and Pain, 2000; Futagi et al., 2008; Park et al., 2008), hence the performance of the algorithm depends on the coverage of such resources.

- L1-specific approaches are shown to be effective (Chang et al., 2008; Dahlmeier and Ng, 2011a), but usually focus on one L1 only. If the learner data is produced by learners with various L1s as in the data in the CLC, this approach is less effective.

- Most previous approaches have either focused on error correction assuming that errors are already detected (Dahlmeier and Ng, 2011a), or performed writing improvement rather than error detection and correction proper (Shei and Pain, 2000; Chang et al., 2008; Futagi et al., 2008). Most of these approaches merge error detection and correction: a content word combination is considered to be an error if there is a more fluent alternative. From the theoretical point of view, this means that "correctness" is viewed as a relative rather than an absolute value. The concept of an "error" is not clearly defined and ED is not performed as an independent step, being mainly integrated into error correction or writing improvement.

- From the practical point of view, since error detection and correction are combined in a single process and since EDC systems rely on manually-created resources for finding alternatives, the system's performance depends on the quality of the corrections found. An originally acceptable combination can be flagged as an error if there is a more fluent alternative, and as a result, the system is prone to false positives. At the same time, a combination that is originally unacceptable can be missed by the EDC system if the system is unable to find an appropriate alternative or score it higher than the original combination. As a result, the EDC system is also prone to false negatives.

- It has been shown that learners often benefit from simply knowing that some combinations need rewriting even if they are not provided with a specific correction (Leacock et al., 2009; Andersen et al., 2013). Since ED is not performed as a separate step, a system based on this approach cannot provide a learner with useful information about the potential error if it has not found some possible correction.

- Such ED systems cannot be effectively applied to previously unseen combinations: the measures used for assessing fluency of the original content word combinations and comparing them to the alternatives are based on frequency of occurrence. Therefore, *any* alternative for the originally unseen content word combination will be suggested by the system as a correction if it is attested in the corpus of English. If such a "correction" is found, the original combination will be flagged as an error, even if the original one is correct (for example, in the AN dataset described in Chapter 3, 78.89% of the previously unseen combinations are correct OOC, and 50.84% are correct IC). At the same time, if no appropriate alternative is found for the original combination, the system cannot make an informed decision about whether the combination is an error or not.

4.2 Experimental setting

For the baseline system, we replicate the three-step algorithm described in §2.3.2.

Detection of miscollocations

We do not use reference databases for collocations and miscollocations as in some previous approaches, and do not rely on frequency of occurrence as evidence for correctness or incorrectness. We assume that a certain number of content word combinations would not be seen even in a relatively big corpus, but would still be correct and semantically acceptable. We apply the ED algorithm to all AN and VO combinations in our datasets.

Search for alternatives

A set of alternatives is created for each component word within the combinations. As with most previous approaches, we primarily consider semantically related words as possible alternatives. The alternatives are extracted from *WordNet 3.0*, and the following WordNet relations are explored:

- *nouns*: synonyms, hyper-/hyponyms;
- *verbs*: synonyms, hyper-/hyponyms;
- *adjectives*: synonyms, *similar terms.*

The set of *similar terms* returns adjectives related semantically to the given one beyond its synonymy set. For example, for *usual*, in addition to the synonym *common*, this relation returns *familiar*, *habitual* and *regular*, among others, while for *big*, in addition to the synonyms *large* and *great*, it returns other adjectives including *important*, *huge*, *wide* and *broad*. Such related adjectives may cover many of the terms that learners find confusing.

The alternative combinations are created by crossing the related words from the sets of alternatives for the component words. Some previous approaches have distinguished between focal words (nouns in both ANs and VOs) and collocates (adjectives in ANs and verbs in VOs), and only consider alternatives for the collocates as those are more frequently chosen incorrectly. Our analysis shows that collocate words are, indeed, more frequently misused than focal words (see Table 3.9), yet the number of cases when the focal words are incorrectly used is also high. We run two sets of experiments, considering alternatives for both words as well as for collocate words only, and compare the results.

Selection of the correction

We compare the original word combination to the alternatives using the *normalised pointwise mutual information* (*NPMI*) score:

$$npmi(ab) = \frac{pmi(ab)}{-log(p(ab))} \tag{4.1}$$

where

$$pmi(ab) = log\frac{p(ab)}{p(a)p(b)} \tag{4.2}$$

and $p(ab)$ is a joint probability of the words a and b occurring in a combination. The probabilities are estimated using maximum likelihood estimation on the basis of word occurrences in a native corpus of English. We have tested this approach with three reference corpora: the BNC, the ukWaC, and a combination of the two, to check how the results are affected by the choice of a reference corpus. Normalisation of the PMI score puts the value in the range of $[-1, +1]$, where -1 is used for words never co-occurring within a combination, 0 is for independent words, and +1 denotes complete co-occurrence.

If some alternative has a higher *NPMI* score than the original combination, the original word combination is considered to be an error and the alternative is suggested as its correction.

4.3 Results

4.3.1 AN combinations

We create the alternatives for AN combinations using four settings:

- 1: adjectives={*original, synonyms*} × nouns={*original, synonyms*};
- 2: adjectives={*original, synonyms, similar*} × nouns={*original, synonyms*};
- 3: adjectives={*original, synonyms*} × nouns={*original, synonyms, hyper-/hyponyms*};
- 4: adjectives={*original, synonyms, similar*} × nouns={*original, synonyms, hyper-/hyponyms*}.

The settings are ordered by the power of the sets of alternatives used: under setting 4 we consider a much wider set of alternatives than under setting 1. Another aspect that is captured by these settings is semantic similarity of the alternatives considered: the wider sets of alternatives contain both semantically close and more semantically distant words, while the smaller sets contain only the closely related words. The four settings allow us to compare how this affects the performance of the system.

We use three reference corpora to estimate the *NPMI* values (see §4.1).

We also check if the assumption that the collocate word is more often incorrectly chosen than the focal word has an effect on the results: we run the first set of experiments considering the alternatives for both words, and the second set of experiments assuming that the noun is chosen correctly and considering the alternatives for the adjectives only.

CLC-FCE dataset

The CLC-FCE dataset is highly skewed: out of the total of $5,211$ ANs, $4,681$ are correct and only 530 are incorrect. This results in a high majority baseline of 0.8983, with the correct instances being the majority class.

Tables 4.1 to 4.4 report our results. Each table reports the results with both words being considered as possibly incorrect (*both:* in the first column of the table), or only adjectives

Setting	BNC	ukWac	BNC+ukWaC
both:1	**0.5020**	0.4504	0.4496
both:2	0.3615	0.3260	0.3280
both:3	0.2998	0.2875	0.2781
both:4	0.2355	0.2218	0.2199
adj:1	**0.6728**	0.6327	0.6339
adj:2	0.4809	0.4348	0.4364

Table 4.1: Detection accuracy, ANs, CLC-FCE dataset.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.9278	**0.4832**	**0.7055**	0.9324	0.4185	0.6755	0.9344	0.4166	0.6755
both:2	0.9390	0.3093	0.6242	0.9527	0.2628	0.6078	0.9531	0.2649	0.6090
both:3	0.9403	0.2354	0.5878	0.9507	0.2181	0.5844	0.9518	0.2068	0.5793
both:4	0.9474	0.1577	0.5525	**0.9630**	0.1391	0.5510	0.9597	0.1374	0.5485
adj:1	0.9122	**0.7035**	**0.8078**	0.9212	0.6464	0.7838	0.9210	0.6479	0.7845
adj:2	0.9255	0.4591	0.6923	**0.9411**	0.3956	0.6683	0.9400	0.3980	0.6690

Table 4.2: System performance on correct AN combinations, CLC-FCE dataset.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	**0.1277**	0.6679	0.3978	0.1248	0.7321	0.4284	0.1258	0.7415	0.4337
both:2	0.1188	0.8226	0.4707	0.1196	0.8849	0.5023	0.1199	0.8849	0.5024
both:3	0.1139	0.8679	0.4909	0.1153	0.9000	0.5077	0.1147	0.9075	0.5111
both:4	0.1103	0.9226	0.5165	0.1114	**0.9528**	**0.5321**	0.1108	0.9491	0.5299
adj:1	0.1330	0.4019	0.2675	0.1407	0.5113	0.3260	**0.1408**	0.5094	0.3251
adj:2	0.1236	0.6736	0.3986	0.1277	**0.7811**	**0.4544**	0.1273	0.7755	0.4514

Table 4.3: System performance on incorrect AN combinations, CLC-FCE dataset.

(*adj:*). The sets of alternatives are also specified: for example, *both:1* denotes setting 1 when the sets of alternatives for both adjectives and nouns are augmented with synonyms, while *adj:1* denotes setting 1 when the set of alternatives is augmented with synonyms for the adjectives only. The best results are marked in bold across the different settings under the condition *both* as well as under the condition *adj*.

Table 4.1 reports the general detection accuracy. It shows how often the algorithm correctly identifies a correct combination as correct and an incorrect combination as incorrect. The exact correction (the combination that is chosen by the algorithm as the most fluent one) at this point is not taken into account: if the original combination X is incorrect and the gold standard correction is Y while the algorithm suggests Z this is still counted as a hit since the error is detected.

Table 4.1 shows that the highest accuracy for both conditions is achieved when both the smaller corpus (BNC) and the smaller set of alternatives (synonyms only) are considered. Since the number of originally correct combinations is high, the system is prone to overcorrection. When it is presented with a wider set of alternatives (settings 2 to 4) or bigger

Setting	BNC	ukWac	BNC+ukWaC
both:1	**0.4377**	0.3798	0.3780
both:2	0.2815	0.2393	0.2412
both:3	0.2138	0.1992	0.1886
both:4	0.1434	0.1274	0.1257
adj:1	**0.6358**	0.5855	0.5868
adj:2	0.4168	0.3608	0.3627

Table 4.4: Correction accuracy, ANs, CLC-FCE dataset.

reference corpora (ukWaC or the combined corpus) it has a wider choice and "corrects" more. The results with the BNC only are also better than those with the other corpora. Accuracy is higher when the algorithm considers only collocate words (cf. 0.5020 and 0.6728): this can be explained by the assumption that adjectives are more frequently incorrectly used, but also by the fact that under the *adj* condition the algorithm considers a smaller set of alternatives than for *both*. We note that the best result of 0.6728 obtained with the *adj* condition and the BNC as a reference corpus is still significantly lower than the baseline of 0.8983.

Tables 4.2 and 4.3 report the precision, recall and F$_1$-measure values on the subsets of correct and incorrect ANs. The results on these subsets are complementary: higher precision in detecting incorrect instances, as well as higher recall and F$_1$-measure on the correct instances, are obtained with a smaller set of alternatives (setting 1 for *both* and *adj*), while higher precision on the correct instances and higher recall and F$_1$-measure on the incorrect instances are obtained with a wider set of alternatives (setting 4 for *both* and *adj*). This shows that when the algorithm is presented with a wider set of alternatives it "finds" more errors and overcorrects, which results in higher recall but lower precision on the incorrect instances. In contrast, when it is presented with a smaller set of alternatives, it corrects less and shows lower recall but higher precision on the incorrect instances.

We also estimate accuracy with respect to the corrections suggested by the algorithm. The results are presented in Table 4.4. Correction accuracy is lower than detection accuracy, but we note the same pattern: the best results are obtained with the smallest set of alternatives consisting of synonyms for adjectives and using the BNC. Correction accuracy is about 3%-4% lower than detection accuracy, but this is due to the fact that the incorrect ANs for which corrections are considered constitute only about 10% of the dataset.

Annotated AN dataset

We first report the results for the OOC annotation, where the baseline is 0.7889, with the correct combinations being the majority class (see Table 3.7). Tables 4.5 to 4.7 report the results for OOC annotation. Tables 4.8 to 4.10 report the results for IC annotation, where the majority baseline is 0.5084, with the correct combinations being the majority class.

Tables 4.5 and 4.8 report the accuracy of the ED algorithm. The results show that the accuracy is substantially lower than the majority baseline for OOC annotation: 0.3897 if both words are allowed to be changed, and 0.5284 if only adjectives are considered

Setting	BNC	ukWac	BNC+ukWaC
both:1	**0.3810**	0.3383	0.3271
both:2	0.2757	0.2481	0.2431
both:3	0.2619	0.2531	0.2506
both:4	0.2368	0.2281	0.2268
adj:1	**0.5313**	0.4574	0.4474
adj:2	0.3358	0.3008	0.2895

Table 4.5: Detection accuracy, ANs, OOC annotation.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.7860	**0.3017**	**0.5439**	**0.7901**	0.2259	0.5080	0.7857	0.2085	0.4971
both:2	0.7236	0.1406	0.4321	0.7200	0.0853	0.4027	0.7042	0.0790	0.3916
both:3	0.7683	0.0995	0.4339	0.7534	0.0869	0.4202	0.7536	0.0822	0.4179
both:4	0.7069	0.0648	0.3858	0.6809	0.0506	0.3657	0.6818	0.0474	0.3646
adj:1	0.8076	**0.5371**	**0.6724**	**0.8145**	0.4092	0.6118	0.8137	0.3934	0.6035
adj:2	0.7588	0.2385	0.4987	0.8000	0.1580	0.4790	0.7797	0.1453	0.4625

Table 4.6: System performance on correct AN combinations, OOC annotation.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.2036	0.6848	0.4442	**0.2058**	0.7697	0.4878	0.2048	0.7818	0.4933
both:2	0.1941	0.7939	0.4940	0.1992	0.8727	0.5359	0.1981	0.8727	0.5354
both:3	0.2039	0.8848	0.5444	0.2028	0.8909	0.5468	0.2030	0.8970	0.5500
both:4	0.2000	0.8970	0.5485	0.1997	0.9091	0.5544	0.2003	**0.9152**	**0.5577**
adj:1	**0.2228**	0.5091	0.3660	0.2208	0.6424	0.4316	0.2195	0.6545	0.4370
adj:2	0.1953	0.7091	0.4522	0.2080	**0.8485**	**0.5283**	0.2044	0.8424	0.5234

Table 4.7: System performance on incorrect AN combinations, OOC annotation.

versus the baseline of 0.7889. As the original combinations are not attested in the BNC and have zero or very low counts in the ukWaC, any alternative that is attested in the corpus is chosen as a correction, resulting in overcorrection. Since the baseline for IC annotation is lower, the difference between the algorithm's accuracy and the baseline is smaller: 0.5378 versus the baseline of 0.5084. In all experiments, the best-performing system uses only synonyms for alternatives, while performance drops steadily as more alternatives are included in the sets. For most experiments, the results are higher when the BNC is used for assessing the "correctness" of the combinations. This is in accord with the observations and results obtained on the CLC-FCE dataset.

Next, we analyse the system's performance on the subsets of correct and incorrect combinations. Precision, recall and F$_1$-measure on the correct combinations are reported in Tables 4.6 and 4.9, and those for the incorrect combinations in Tables 4.7 and 4.10. The highest precision, recall and F$_1$-measure on the subset of correct OOC combinations are achieved with the smaller set of alternatives, which corresponds to the results obtained for accuracy and can be explained by the fact that the system tends to overcorrect when

Setting	BNC	ukWac	BNC+ukWaC
both:1	0.4449	**0.4624**	0.4536
both:2	0.4236	0.4386	0.4361
both:3	0.4398	0.4474	0.4449
both:4	0.4323	0.4273	0.4286
adj:1	**0.4937**	0.4724	0.4699
adj:2	0.4549	0.4486	0.4424

Table 4.8: Detection accuracy, ANs, IC annotation.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.4979	**0.3048**	0.4014	0.5635	0.2569	**0.4102**	0.5476	0.2317	0.3897
both:2	0.4634	0.1436	0.3035	0.5733	0.1083	0.3408	0.5634	0.1008	0.3321
both:3	0.5732	0.1184	0.3458	**0.6438**	0.1184	0.3811	0.6377	0.1108	0.3743
both:4	0.5690	0.0831	0.3260	0.5532	0.0655	0.3093	0.5682	0.0630	0.3156
adj:1	0.5249	**0.5567**	**0.5408**	0.5283	0.4232	0.4757	0.5261	0.4055	0.4658
adj:2	0.5176	0.2594	0.3885	**0.5440**	0.1713	0.3576	0.5254	0.1562	0.3408

Table 4.9: System performance on correct AN combinations, IC annotation.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.5027	0.6958	0.5992	**0.5219**	0.8030	0.6624	0.5159	0.8105	0.6632
both:2	0.4963	0.8354	0.6659	0.5104	0.9202	0.7153	0.5089	0.9227	0.7158
both:3	0.5112	0.9127	0.7119	0.5172	0.9352	0.7262	0.5158	0.9377	0.7267
both:4	0.5081	0.9377	0.7229	0.5060	0.9476	0.7268	0.5066	**0.9526**	**0.7296**
adj:1	**0.5332**	0.5012	0.5172	0.5229	0.6259	0.5744	0.5203	0.6384	0.5794
adj:2	0.5092	0.7606	0.6349	0.5111	0.8579	**0.6845**	0.5074	**0.8603**	0.6839

Table 4.10: System performance on incorrect AN combinations, IC annotation.

wider sets of alternatives are considered. For correct IC combinations the highest recall and F$_1$-measure are achieved with the smaller set of alternatives, while the best precision values are achieved with wider sets of alternatives. This pattern is similar to the results obtained for correct combinations in the CLC-FCE dataset (see Table 4.2): since error annotation in the CLC-FCE dataset is provided for use in context as well, we would expect to see some correspondence between the results.

Tables 4.7 and 4.10 report the results on the incorrect instances and show that the highest precision is also achieved using the smaller set of alternatives. Highest recall and F$_1$-measure are achieved using the widest set of alternatives and also using either ukWaC or a combination of the two corpora. This shows that the system has higher coverage when it considers more alternatives. Since it is prone to overcorrection, precision values on the incorrect instances are quite low: precision of 0.2406 on the OOC annotation means that only about every fourth "error" identified by this system is an actual error, while precision of 0.5293 on the IC means that only about half of the identified "errors" are actual errors.

Comparison of the results obtained when the error location is fixed to be on the adjective

Setting	BNC	ukWac	BNC+ukWaC
both:1	**0.1842**	0.1591	0.1491
both:2	0.1103	0.0940	0.0915
both:3	0.0664	0.0689	0.0639
both:4	0.0476	0.0414	0.0401
adj:1	**0.3158**	0.2481	0.2393
adj:2	0.1855	0.1466	0.1391

Table 4.11: Correction accuracy, ANs.

with those obtained when both words are considered shows that the system is substantially more accurate in ED on OOC annotation (cf. 0.3897 and 0.5284 in Table 4.5), though the results are not significantly different for IC annotation (Table 4.8). The system gains 2%-3% precision on OOC annotation (Tables 4.6 and 4.7), but performs about the same or worse in terms of precision on IC annotation (Tables 4.9 and 4.10).

We also check how often the system correction coincides with the one suggested by the annotators. Table 4.11 shows that the best result of 0.3319 is obtained when only adjectives and their synonyms are considered. There is a substantial drop of about 20% as compared to detection accuracy of the algorithm (cf. results in Tables 4.5 and 4.8).

4.3.2 VO combinations

We create the alternatives for VO combinations using two settings:

- 1: verbs=$\{$*original, synonyms*$\}$ × nouns=$\{$*original, synonyms*$\}$;
- 2: verbs=$\{$*original, synonyms, hyper-/hyponyms*$\}$ × nouns=$\{$*original, synonyms, hyper-/hyponyms*$\}$;

We follow the procedure applied to the ANs, testing the effect of the set of alternatives as well as the effect of the reference corpus on the results. The set of alternatives under setting 1 contains more closely related words than under setting 2. We run two sets of experiments: considering the alternatives for both words within a combination (*both*), and considering only verbs as possibly incorrectly chosen (*verbs*).

CLC-FCE dataset

The VO subset is also highly skewed towards correct combinations: out of the total of $5,700$ VOs, $4,911$ are correct and only 789 are incorrect. This results in a high majority baseline of 0.8616.

The results are reported in Tables 4.12 to 4.15, using the same procedure as with the ANs.

We obtain higher detection accuracy (Table 4.12) with the smaller set of alternatives (setting 1), smaller reference corpus (BNC only), and when the error location is fixed to

Setting	BNC	ukWac	BNC+ukWaC
both:1	**0.3618**	0.3402	0.3395
both:2	0.1911	0.1882	0.1882
verb:1	**0.4863**	0.4558	0.4568
verb:2	0.3009	0.3039	0.3037

Table 4.12: Detection accuracy, VOs, CLC-FCE dataset.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.9046	**0.2898**	**0.5972**	0.9272	0.2541	0.5907	0.9276	0.2531	0.5904
both:2	0.9076	0.0680	0.4878	**0.9356**	0.0621	0.4988	0.9329	0.0623	0.4976
verb:1	0.8984	**0.4553**	**0.6768**	0.9140	0.4066	0.6603	0.9165	0.4066	0.6616
verb:2	0.9012	0.2118	0.5565	0.9338	0.2067	0.5702	**0.9345**	0.2063	0.5704

Table 4.13: System performance on correct VO combinations, CLC-FCE dataset.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.1548	0.8099	0.4824	**0.1587**	0.8758	0.5172	**0.1587**	0.8771	0.5179
both:2	0.1416	0.9569	0.5493	0.1429	**0.9734**	**0.5581**	0.1428	0.9721	0.5574
verb:1	0.1669	0.6793	0.4231	0.1710	0.7617	0.4664	**0.1724**	0.7693	0.4709
verb:2	0.1485	0.8555	0.5020	0.1554	0.9087	0.5321	0.1555	**0.9100**	**0.5328**

Table 4.14: System performance on incorrect VO combinations, CLC-FCE dataset.

Setting	BNC	ukWac	BNC+ukWaC
both:1	**0.2546**	0.2263	0.2251
both:2	0.0609	0.0563	0.0563
verb:1	**0.3991**	0.3593	0.3591
verb:2	0.1893	0.1867	0.1863

Table 4.15: Correction accuracy, VOs, CLC-FCE dataset.

be on the verb. Again, we assume that since verbs are more frequently incorrectly chosen than nouns, a system that considers alternatives for verbs only might be expected to perform better. Fixing the error location on one component within the combination also reduces the set of alternatives and the system becomes less prone to overcorrection. The best detection accuracy of 0.4863 is still substantially lower than the baseline of 0.8616.

We analyse the performance of the system on the correct and incorrect instances, and report the results in Tables 4.13 and 4.14. As before, we observe that the results for the correct and incorrect instances are complementary: the system identifies more "errors" when it is presented with a wider set of alternatives and uses a bigger corpus for estimating fluency. Since under such conditions it identifies many "errors", this results in the highest recall and F_1-measure for the incorrect combinations and also in high precision on the correct combinations – see *both:2* rows in the Tables 4.13 and 4.14. But since this results in overcorrection, precision on the incorrect instances is higher when a smaller set of alternatives is considered. However, even under this setting precision on the incorrect

Setting	BNC	ukWac	BNC+ukWaC
both:1	0.4918	**0.4969**	0.4881
both:2	0.4580	0.4592	0.4580
verb:1	**0.5220**	0.5169	0.5056
verb:2	0.4768	0.4617	0.4617

Table 4.16: Detection accuracy, VOs, OOC annotation.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.6175	**0.2522**	0.4349	0.6975	0.1853	**0.4414**	0.6786	0.1696	0.4241
both:2	0.6905	0.0647	0.3776	**0.7576**	0.0558	0.4067	0.7500	0.0536	0.4018
verb:1	0.6184	**0.3906**	**0.5045**	0.6567	0.2946	0.4757	0.6378	0.2790	0.4584
verb:2	0.6598	0.1429	0.4013	0.6508	0.0915	0.3712	**0.6610**	0.0871	0.3740

Table 4.17: System performance on correct VO combinations, OOC annotation.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.4544	0.7994	0.6269	**0.4617**	0.8968	0.6793	0.4569	0.8968	0.6769
both:2	0.4450	0.9628	0.7039	0.4463	**0.9771**	**0.7117**	0.4458	**0.9771**	0.7114
verb:1	0.4689	0.6905	0.5797	**0.4698**	0.8023	0.6360	0.4626	0.7966	0.6296
verb:2	0.4514	0.9054	0.6784	0.4455	0.9370	0.6912	0.4458	**0.9427**	**0.6942**

Table 4.18: System performance on incorrect VO combinations, OOC annotation.

instances is quite low: precision of 0.1724 means that only about every fifth "error" identified by this algorithm is an actual error.

Correction accuracy reported in Table 4.15 is about 9% to 11% lower than detection accuracy. The drop in accuracy is bigger than what we observe for ANs (see Table 4.4) showing that the corrections selected by the system from the set of synonyms coincide with the gold standard correction less often than for the AN combinations.

Annotated VO dataset

We report the results for OOC annotation (Tables 4.16 to 4.18) and for IC annotation (Tables 4.19 to 4.21). The baseline for OOC annotation is 0.5557 with the correct instances being the majority class, and 0.6086 for IC annotation with the incorrect instances being the majority class.

Detection accuracy is reported in Tables 4.16 and 4.19. It shows a slightly different pattern than before, for the AN combinations and the CLC-FCE VO subset. The best results, as before, are obtained with the smaller set of alternatives, and accuracy for OOC annotation is higher when only verbs are considered. However, the absolute difference in accuracy is only about 3%, and the system performs better on IC annotation when alternatives for both words are considered. The results are also higher with the ukWaC used as a reference corpus. None of the results, however, beats the majority baseline.

Setting	BNC	ukWac	BNC+ukWaC
both:1	0.5784	**0.6048**	0.5960
both:2	0.5797	0.5847	0.5834
verb:1	0.5646	**0.5997**	0.5885
verb:2	0.5797	0.5809	0.5834

Table 4.19: Detection accuracy, VOs, IC annotation.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.4973	**0.2926**	0.3949	0.6134	0.2347	**0.4241**	0.5893	0.2122	0.4008
both:2	0.6190	0.0836	0.3513	**0.7273**	0.0772	0.4022	0.7188	0.0740	0.3964
verb:1	0.4700	**0.4277**	0.4488	0.5473	0.3537	**0.4505**	0.5255	0.3312	0.4283
verb:2	0.5155	0.1608	0.3381	0.5873	0.1190	0.3531	**0.6102**	0.1158	0.3630

Table 4.20: System performance on correct VO combinations, IC annotation.

Setting	BNC			ukWac			BNC+ukWaC		
	P	R	F_1	P	R	F_1	P	R	F_1
both:1	0.6417	0.8107	0.7262	**0.6490**	0.9053	0.7772	0.6423	0.9053	0.7738
both:2	0.6225	0.9671	0.7948	0.6243	**0.9815**	**0.8029**	0.6235	**0.9815**	0.8025
verb:1	0.6537	0.6914	0.6725	**0.6628**	0.8128	0.7378	0.6539	0.8086	0.7313
verb:2	0.6271	0.9033	0.7652	0.6267	0.9465	0.7866	0.6274	**0.9527**	**0.7900**

Table 4.21: System performance on incorrect VO combinations, IC annotation.

Setting	BNC	ukWac	BNC+ukWaC
both:1	**0.1330**	0.1142	0.1029
both:2	0.0414	0.0376	0.0389
verb:1	**0.1932**	0.1681	0.1581
verb:2	0.0903	0.0728	0.0715

Table 4.22: Correction accuracy, VOs.

The system's performance on correct instances is presented in Tables 4.17 and 4.20, while that on incorrect instances is in Tables 4.18 and 4.21. The results follow the same pattern as before: the smaller set of alternatives results in higher precision on the incorrect instances and higher recall and F_1-measure on the correct VOs, while a wider set of alternatives results in higher precision on the correct VOs and higher recall and F_1-measure on the incorrect ones. Precision on the incorrect VOs with the OOC annotation is under 50% (P=0.4698, Table 4.18) which means that the system is more often wrong about errors detected than it is right. However, precision rises to 0.6628 (Table 4.21) for the IC annotation: since 60.86% of the VOs are incorrect in context, a system that identifies many "errors" can be expected to perform better under such conditions.

Correction accuracy shows a significant drop from detection accuracy, which shows that most of the time the corrections returned by the system are different from the gold standard ones. This is possibly due to the fact that the proportion of the semantically related confusions in this dataset is generally only about one-fourth (see Table 3.10). This sets the upper bound on correction accuracy for the system that only considers semantically related confusions at 0.2587.

4.4 Analysis and discussion

We analyse the results obtained in our experiments with the baseline system and compare them to the observations outlined in §4.1:

- The system is primarily concerned with finding an alternative with the highest collocational strength rather than with correcting word combinations. As a result, it is prone to overcorrection and returns a high number of false positives, which makes the system unreliable in practice. For example, it suggests correcting *funny actor* to *comic actor*, *short speech* to *short address*, and *see parent* to *visit parent*: the original combinations are correct but have lower collocational strength than their alternatives. This results in lower precision on the incorrect combinations (0.1277-0.1408 for the ANs, and 0.2058-0.2228 for the VOs), as well as in low accuracy. The problem of the high number of false positives can be partly solved by fixing the error location on the collocating word: in our experiments, this results in an improvement in performance, but accuracy is still lower than the majority class baseline.
  The results might be expected to change if one uses another metric for measuring collocational strength. However, the metrics that are based on frequency of occurrence will fail to detect errors in the controlled annotated dataset. The results of the baseline system on this dataset are very low and unlikely to be improved with any modifications on the component parts of the system. The accuracy of ED in previously unseen combinations is only 0.3810 for the OOC and 0.4624 for the IC annotation on ANs, and 0.4969 for the OOC and 0.6048 for the IC annotation on VOs, if both words within word combinations are considered. Since the original combinations are not seen in the corpus, *any* corpus-attested alternative can be selected as a correction for the original combination by the baseline algorithm: for example, the system corrects *important conversation* to *serious conversation*, *big examination* to *big test*, *attend speech* to *attend lecture* and *obtain tuition* to *receive tuition*.

- The system combines error detection and correction and makes the detection step dependent on the correction and on the set of alternatives found by the system. In this implementation, we only considered semantically related words as possible alternatives. Expanding the set of alternatives can improve the quality of the system's suggestions, but our experiments also show that when the system is presented with a wider set of alternatives it tends to overcorrect more, so the best accuracy is achieved when the smallest set is considered.
  One of the weaknesses of this system is that it does not take the original meaning into account: so, for example, a *comic actor* is more fluent than a *funny actor* but does not mean the same thing.

- It is important to make sure that the system finds a representative set of alternatives which, at the same time, are sufficiently similar to the original words. WordNet

provides the latter but does not cover all possible corrections. As a result, in addition to the false positives, the system also returns a substantial number of false negatives: some errors are not detected either because the system has not found appropriate alternatives, as, for example, for **high shyness* and **effect area*, or has failed to score them higher than the original combination.
It also means that some originally correct combinations are not falsely identified by this system as errors not because the system is good at ED, but simply because no possible alternatives are found. In order to give a more realistic estimation of the system's performance, we exclude the examples for which the system makes no comparison to the alternatives, reducing the set of previously unseen ANs from 798 to 401, and re-evaluate the system: for the ANs annotated OOC, the best accuracy drops from 0.4863 to 0.4185, and recall on the correct instances drops from 0.5371 to 0.3318. For IC annotation, the best accuracy drops from 0.4937 to 0.4712, and recall on the correct combinations drops from 0.5567 to 0.3501. Similarly, if we exclude the VOs for which the system makes no comparison to the alternatives, we will reduce the set of test VOs from 800 to 518. The best accuracy for the VO OOC annotation will drop from 0.5220 to 0.4981, and the recall on the correct instances from 0.3906 to 0.2277. For the IC annotation, the best accuracy will drop from 0.6048 to 0.5922, and the recall on the correct combinations from 0.4277 to 0.2797.

- The system is implemented so that it chooses one correction only. As a result, the system might not be rewarded for the alternative acceptable corrections if they are not included in the gold standard: for example, the system corrects **make damage* to *cause damage* while the gold standard suggests *do damage* as the correction. Both corrections are acceptable, but if the system is evaluated using only a limited set of corrections, its performance will inevitably be underestimated (see the discussion in §2.2). We report both detection and correction accuracy, where the former shows how accurate the system is if the specific correction by the system is not taken into account. In a number of cases where the system detects an error it does not suggest an appropriate correction: for example, it suggests correcting **open TV* to *afford TV* while the gold standard correction is *turn on TV*.

We note that the system still helps in identifying a certain number of errors: for example, it detects *big knowledge* and suggests *great knowledge* as a correction. It also identifies errors in more trivial cases: for example, it suggests correcting *economical position* to *economic situation* and *elder people* to *older people*. Nevertheless, due to the limitations outlined above, the baseline system is mostly inappropriate for handling learner errors.

Chapter 5

Semantic models for error detection

In Chapter 2, we maintained that compositional distributional semantic models should be applied to ED in content words since many errors are caused by a semantic mismatch between the words chosen (for example, *deep majesty*, *economic child* or *electric society*). In this chapter, we discuss our experiments with the semantic models.

We follow the implementation procedure described in Vecchi et al. (2011). The semantic space construction is discussed in §5.1, and the experimental setting in §5.2. In addition to the measures of semantic anomaly detection originally proposed by Vecchi et al. (2011), we introduce other ones that we assume can reliably detect the difference between the vectors for the correct and incorrect combinations. These measures are discussed in §5.3.

The results of the experiments on the AN combinations are presented in §5.4, and the application of the models of compositional distributional semantics to VO combinations is discussed in §5.5 to §5.8. We conclude with §5.9, where we also outline which of the models and semantic measures perform best and can be used as features by an ED algorithm.

5.1 Semantic space construction

5.1.1 Original semantic space

The *semantic space* for the ANs is populated with a large number of distributional vectors representing the meaning of the *target elements* – constituent nouns and adjectives from the test ANs and the most frequent nouns and adjectives from a corpus of English, as well as combinations of these nouns and adjectives. The distributional vectors for the input words are used to generate compositional vectors for the test ANs. In addition to the adjectives and nouns that are contained in the test ANs, we also need the distributional vectors for the frequent nouns and adjectives which are not necessarily part of the test ANs, and for the combinations containing these frequent nouns and adjectives, since a number of measures that we apply to distinguish between acceptable and deviant AN vectors are based on exploration of the semantic space and neighbourhood of the model-generated vectors (see §5.3). To estimate the frequency rankings and collect the *vocabulary*

	$V = v$	$V \neq v$	
$U = u$	O_{11}	O_{12}	$= R_1$
$U \neq u$	O_{21}	O_{22}	$= R_2$
	$= C1$	$= C_2$	$= N$

Table 5.1: Observed frequencies.

	$V = v$	$V \neq v$
$U = u$	$E_{11} = \frac{R_1C_1}{N}$	$E_{12} = \frac{R_1C_2}{N}$
$U \neq u$	$E_{21} = \frac{R_2C_1}{N}$	$E_{22} = \frac{R_2C_2}{N}$

Table 5.2: Expected frequencies.

of the most frequent adjectives and nouns, we use a concatenation of two well-formed English corpora – the $100M$-word BNC and the Web-derived $2B$-word ukWaC corpus.[1]

The semantic space is represented by a matrix encoding word co-occurrences, with the rows representing the target elements and the columns representing a set of $10K$ *context words* consisting of the most frequent $6,590$ nouns, $1,550$ adjectives and $1,860$ verbs in the combined corpus. The ij-th cell of the original matrix contains a sentence-internal co-occurrence count of the i-th target element with the j-th context word. The raw sentence-internal co-occurrence counts from the original matrix are transformed into *LMI* scores as used in Baroni and Zamparelli (2010) and Vecchi et al. (2011), and originally proposed by Evert (2005). *LMI* is a variation of the mutual information measure which takes into account the observed co-occurrence counts of the two words which are not necessarily dependent on each other. It is estimated as a product of the observed co-occurrence count and mutual information score, where the *MI* score indicates how much information the components of the word pair provide about each other *in general*, i.e. averaged over all pair types in the population (Evert, 2005, p. 89). *LMI* scores are estimated as:

$$LMI = O_{11} \cdot log\frac{O_{11}}{E_{11}} \tag{5.1}$$

O_{11} and E_{11} correspond to the observed and expected counts for the word pair, respectively, which can be estimated from the contingency table. Tables 5.1 and 5.2 illustrate how O_{11} and E_{11} can be estimated for a pair of words U and V. Simply put, O_{11} denotes the co-occurrence count for the pair of words $count(uv)$, while E_{11} denotes $\frac{count(u*) \cdot count(*v)}{N}$ where $count(u*)$ is the number of bigrams starting with u and $count(*v)$ is the number of bigrams ending with v.

LMI is a direct extension of the commonly used *mutual information* (*MI*) measure and the two measures can be linked as:

$$\begin{aligned} LMI \quad = O_{11} \cdot log\tfrac{O_{11}}{E_{11}} \quad &= count(uv) \cdot log(count(uv) / \tfrac{count(u*) \cdot count(*v)}{N}) \\ &= count(uv) \cdot log(\tfrac{count(uv)}{N} / \tfrac{count(u*) \cdot count(*v)}{N \cdot N}) \\ &= count(uv) \cdot log(\tfrac{count(uv)}{N} / (\tfrac{count(u*)}{N} \cdot \tfrac{count(*v)}{N})) \\ &= count(uv) \cdot log\tfrac{P(uv)}{P(u*) \cdot P(*v)} \\ &= count(uv) \cdot MI \end{aligned} \tag{5.2}$$

Baroni and Zamparelli (2010) and Vecchi et al. (2011) report that *LMI* is an association measure that closely approximates the commonly used *Log-Likelihood Ratio*, but is simpler to compute.

[1] `http://wacky.sslmit.unibo.it/`

We have set the semantic space using frequency rankings from the concatenated corpus consisting of the BNC and the ukWaC. This allows us to explore the effect of the input corpus in estimating co-occurrence counts. In this project, we have selected the sets of ANs and VOs unattested in the BNC only (see §3.2). We estimate word co-occurrence statistics using the BNC only as well, and leave it for future research to explore the impact of estimating the co-occurrence counts from a larger corpus. We lemmatise, tag and parse the data with the RASP system (Briscoe et al., 2006; Andersen et al., 2008), and extract all statistics at the lemma level. We use the parsed data to make sure that the adjectives and nouns, as well as the verbs and nouns, in the extracted ANs and VOs are grammatically related, and also to be able to extract the word combinations where the two constituent words are not adjacent.

For the target elements, we first select the $4K$ adjectives and $8K$ nouns which are most frequent in the concatenated corpus. In each case, we exclude the top 50 most frequent words since those may have too general a meaning. Next, we extract the constituent adjectives and nouns from our test data and populate the semantic space with the words not yet contained in it. As a result, our semantic space contains $8,364$ nouns.

Then, we add more AN combinations to the semantic space. We select 218 frequent adjectives (occurring more than $100K$ but less than $740K$ times), merge them with the adjectives from the test ANs, and generate all possible AN combinations by crossing this combined set of adjectives and the set of $8,364$ nouns. This results in a set of ANs of which $1,6M$ combinations are corpus-attested. From these we randomly choose $62,205$ ANs that occur more than 100 times in the corpus. As a result, we populate our semantic space with ANs with the number of unique corpus-attested combinations per adjective ranging from 1 to $1,226$ and being 84.52 on average. Since we apply our approach to real data, we cannot avoid having a different number of training examples for different adjectives. In future it may be worth exploring how many training examples are needed for a single adjective, since some highly frequent adjectives have more training examples.

Finally, we check our test set against the combined corpus and add $1,131$ test ANs which are corpus-attested but not yet contained in the semantic space. Our final semantic space consists of $8,364$ nouns, $4,353$ adjectives and $63,336$ corpus-attested ANs, and is represented by a $76,053 \times 10K$ matrix.

5.1.2 Dimensionality reduction

We apply three models of semantic composition which use distributional vectors from the constructed semantic space. The *add* and *mult* models do not require any additional training, but for the *alm* model we need to estimate a regression model for each dimension and for each test set adjective. Using the original $10K$-dimension distributional vectors has proven to be time-consuming and costly computationally.[2] We applied dimensionality reduction to the original matrix to obtain a more compact space.

We followed the dimensionality reduction procedure outlined in Baroni and Zamparelli (2010), and applied SVD which helps to represent the target words and phrases with their

[2] Vecchi (2013) reports that for certain models, for example the *mult* model, using original non-reduced vectors might improve results.

coordinates in the space spanned by the first n right singular vectors. This technique has been chosen because it not only helps mitigate the dimensionality problem, but it has also been reported to improve the quality of the semantic space (Landauer and Dumais, 1997; Rapp, 2003; Schütze, 1997). SVD is applied to the part of the matrix representing adjectives and nouns, while the AN vectors are projected onto the reduced space by multiplying the original vectors by a matrix containing the first n right singular vectors as columns. Baroni and Zamparelli (2010) have motivated this step by the fact that it helps avoid bias in favour of dimensions that capture variance in the test set ANs. The quality of the reduced matrix has been validated by Baroni and Zamparelli (2010) in an independent set of experiments.

The chosen number of n singular vectors in these experiments is 300, and as a result we get a more compact and dense semantic space represented by a $76,053 \times 300$ matrix.

5.2 Experimental setting

5.2.1 Semantic models

We apply the simple *add* and *mult* models of semantic composition, which derive the AN (an) vectors by component-wise addition and multiplication applied to the adjective (a) and noun (n) vectors:

$$an_i = a_i + n_i \qquad (5.3) \qquad\qquad an_i = a_i \cdot n_i \qquad (5.4)$$

For the *alm* model, the weight coefficients are estimated with multivariate partial least squares regression using the `pls` package (Mevik and Wehrens, 2007) for R (R Core Team, 2014), and applying the leave-one-out training regime. This model is computationally more expensive than the *add* and *mult* models since a separate weight matrix must be learned for each adjective. The number of latent variables used by the training algorithm depends on the number of available noun–AN training pairs. We gradually changed this number from 3 to 40 depending on the adjective and the number of available training pairs, with the aim of keeping the independent-variable-to-training-item ratio stable.

5.2.2 Evaluation procedure

Vecchi et al. (2011) have proposed three measures for detecting semantic anomaly in AN combinations. We introduce 11 additional measures that we hypothesise can help distinguish between correct and incorrect word combinations (see §5.3). The set of model-generated AN vectors is divided into two major groups – the vectors for correct and for incorrect combinations. The measures are applied to the vectors, and the mean values for the correct and incorrect model-generated AN vectors are compared. We use the one-shot unpaired t-tests assuming a two-tailed distribution, and report the results in terms of p

values. If models show a difference for the measures applied to the two groups of vectors at the $p < 0.05$ level, we mark such results in bold when we report them.

The tests show whether the measures can reliably distinguish between the vectors for the correct and incorrect word combinations *in general*. We also assume that the values for the measures can further be used to derive discriminative features for an ML classifier.

5.3 Measures of semantic anomaly

We discuss the measures that are applied to distinguish between the groups of vectors for the correct and incorrect ANs. The measures that have been proposed by Vecchi et al. (2011) are marked with an asterisk ($*$). We group the 14 measures used in our experiments by their type based on the underlying hypotheses:

1. **Measures based on the properties of the model-generated vector**: one of the measures applied to the vectors relies on the hypothesis that the vectors representing correct combinations and those representing incorrect combinations should differ with respect to their length.

2. **Measures based on the relations between the input and the output vectors**: several measures rely on the assumption that vectors representing correct ANs are more closely related to the input vectors than vectors representing incorrect ANs. For example, it is assumed that model-generated vectors for the correct ANs are situated closer in the semantic space to the input noun vectors.

3. **Measures based on the neighbourhood of the model-generated vectors**: a number of measures rely on the hypotheses about how the neighbourhoods for the model-generated vectors in the semantic space look like. We assess the neighbourhoods quantitatively by the density of the neighbourhood, as well as qualitatively by semantic similarity of the neighbours to the model-generated AN vector.

5.3.1 Measures based on the properties of the model-generated vector

- ***Vector length (*VLen*)** – see §2.4 for a description.

5.3.2 Measures based on the relations between the input and the output vectors

These measures rely on comparison of the model-generated (output) vector to the distributional (input) vectors.

- ***Cosine to the input noun (*cosN*)** – see §2.4 for a description.

- **Cosine to the input adjective (*cosA*)** is a measure originally presented in Kochmar and Briscoe (2013). It is analogous to `cosN` measure, but it considers semantic similarity between the model-generated AN vector and the input distributional adjective vector. We assume that not only the noun meaning is 'distorted' in the incorrect ANs, but the meaning of the input adjective is not preserved either. For example, **parliamentary potato* is not semantically related to any parliamentary phenomena, and we hypothesise it should be situated further away in the semantic space from the vector for *parliamentary* than, for example, *parliamentary elections*. Within formal semantics, adjectives are treated as functions mapping from nominal meaning of the input nouns that the adjectives combine with to the nominal meaning of the output ANs. Hence, the nominal meaning plays a more central role than adjectival meaning and the nouns and adjectives do not contribute to the AN meaning in a symmetric manner. However, two of the applied models – the *add* and *mult* – are based on the use of symmetric functions, and it seems justified to apply symmetric measures for semantic anomaly detection to the output of these models.

- **Cosine to the distributional vector for the corpus-attested ANs (*cosAN*).** Some of the incorrect ANs can be corpus-attested (see §3.1). For the corpus-attested ANs, distance from the model-generated vector to the distributional vector can be calculated, and semantically acceptable ANs can be expected to be located closer to their distributional vectors than semantically deviant and non-compositional ones. For example, *parliamentary elections* combines properties of both *parliamentary* events and *elections*. Therefore, the distributional vector and the model-generated vector can be assumed to be close to each other in the semantic space. At the same time, attested but non-compositional phrases will have their model-generated and distributional vectors located further away from each other. For example, the vector derived through composition of the adjective *red* and the noun *herring* will be placed close to *red* objects and *fish* objects, while the distributional vector for *red herring* will be placed in a different part of the semantic space.

5.3.3 Measures based on the neighbourhood of the model-generated vectors

This group of measures is based on quantitative and qualitative analysis of the semantic neighbourhood of the model-generated vectors.

- ***Density of the neighbourhood populated by** 10 **nearest neighbours (*dens*)** – see §2.4 for a description. This measure allows us to quantitatively assess the semantic neighbourhood of the model-generated vectors.

- ***Density among the** 10 **nearest neighbours (*densAll*)** is a modification of the `dens` measure introduced in Vecchi (2013). She hypothesises that model-generated vectors for deviant ANs will share a neighbourhood with elements that are not even similar amongst themselves as they will not inhabit an area of space inhabited by coherent discourse topics. She predicted that ANs with a higher average similarity

between all neighbours would correspond to more acceptable ANs. This is estimated as an average of the 11 density values calculated for each member of the set $\{AN$ *vector*, *its* 10 *nearest neighbours*$\}$, where each density value is estimated as an average distance from the member of the set to all other members of the set.
This measure also allows us to assess the semantic neighbourhood of the model-generated vectors quantitatively, but produces more directly interpretable results.
Vecchi (2013) obtained the results that contradicted the original hypothesis about these measures: a number of models have placed the AN vectors for anomalous combinations in denser neighbourhoods than vectors for acceptable ANs. Qualitative analysis of these neighbourhoods has shown that the vectors for the anomalous ANs have often been "pulled" to the artificially densely populated neighbourhoods overruled by the meaning of the adjective rather than noun. This supports the original hypothesis that it is the meaning of the input noun that should be preserved in the semantically acceptable AN, and the anomalous ANs lose their input noun meaning being surrounded by neighbours similar to the input adjective but not to the noun.

- **Ranked density in close proximity (*Rdens*)** is a measure proposed in Kochmar and Briscoe (2013). It relies on the notion of *close proximity* which is defined as a neighbourhood populated by some very close neighbours. The threshold for close neighbours is set empirically to 0.8 and only neighbours with the cosine equal to or higher than this threshold are considered for estimation of this measure. `RDens` is calculated as $RDens = \sum_{i=1}^{N} rank_i \cdot distance_i$, with N being the total number of close neighbours, each with its rank and distance.
  This measure returns the weighted sum of the distances of the close neighbours, where each distance value is weighted with its rank. The rank is defined so that the closer neighbours' contribution to the sum is weighted more heavily than the more distant neighbours' contribution. Therefore, the AN vectors that have higher number of close neighbours as well as some very close ones with high cosine values get higher `RDens` values. It is assumed that semantically acceptable and correct ANs have more close neighbours and, as a result, score higher.

- **Number of neighbours within close proximity (*num*)**, or N used for the estimation of `RDens`, is used as a separate measure, and we assume that it is lower for incorrect combinations which are expected to be more isolated in semantic space. This measure also assesses the semantic neighbourhood in a quantitative way.

- **Lexical overlap between the 10 nearest neighbours and constituent noun and adjective (*OverAN*)**, introduced in Kochmar and Briscoe (2013), is used to assess semantic neighbourhood qualitatively. We assume that semantically correct ANs should be surrounded by words and combinations similar to the input noun and adjective. This measure is estimated as the proportion of the 10 nearest neighbours containing the same constituent words as in the tested ANs.
  For example, we might expect to see *election* and *parliamentary* among the 10 nearest neighbours for the model-generated vector of *parliamentary elections*, as well as other ANs containing these words. As we assume that **parliamentary potato* is not related to parliamentary phenomena or potatoes, we expect to see lower number of words and ANs coinciding or containing *parliamentary* or *potato*.

- **Lexical overlap between the 10 nearest neighbours and input noun (*OverN*)** is a variant of `OverAN` in which we consider the lexical overlap with the input noun.

- **Lexical overlap between the 10 nearest neighbours and input adjective (*OverA*)** is a variant of `OverAN` in which we consider the overlap with the input adjective.

- **Overlap between the 20 nearest neighbours for the AN and the nearest neighbours for the constituent noun and adjective (*NOverAN*)**, introduced in Kochmar and Briscoe (2014), is another measure that assesses semantic neighbourhood qualitatively. It extends the set of considered neighbours to the nearest neighbours for the input adjective and noun. We have restricted this set by 10 nearest one-word (adjective or noun) neighbours plus 10 nearest two-word (AN) neighbours. We assume that inclusion of both word and phrase neighbours allows us to find a better match between the neighbours of the model-generated vectors and the input distributional vectors. We also believe that this measure is able to go beyond simple lexical overlap and assess the semantic similarity of the neighbours for the input words and their combinations.
  `NOverAN` is estimated as the proportion of the common neighbours among the 20 nearest neighbours for the AN vector and the 20 nearest neighbours for the distributional vectors of the input words. Since we assume that correct ANs and the constituent words should be placed in similar neighbourhoods, we expect to get a higher value for *parliamentary elections* with a higher overlap between its neighbours and the neighbours for *parliamentary* and *elections*, than for **parliamentary potato*.

- **Overlap between the 20 nearest neighbours for the AN and the nearest neighbours for the constituent noun (*NOverN*)** is a variant of `NOverAN` with the neighbours for the input noun considered.

- **Overlap between the 20 nearest neighbours for the AN and the nearest neighbours for the constituent adjective (*NOverA*)** is a variant of `NOverAN` with the neighbours for the input adjective considered.

5.3.4 Tests on corpus-attested word combinations

The CLC-FCE dataset contains $3,294$ correct corpus-attested ANs and 286 corpus-attested ANs that are annotated as incorrect in the learner data. We first run the tests applying the proposed measures to the set of distributional AN vectors for the corpus-attested combinations. Since the distributional vectors are sparse, this may affect the results.

The results are reported in Table 5.3, and the measures that show the difference at the $p < 0.05$ level are marked in bold. Since for all our measures we initially predict higher values on the correct ANs than on the incorrect ones, we mark the results that support this intuition in blue. When we see that the values on the incorrect combinations are higher than those on the correct ones, with $p < 0.05$, we mark these results in red.

Metric	*distributional*
VLen	**8.28**$*10^{-4}$
cosN	0.6625
cosA	0.1394
cosAN	1
dens	0.3316
densAll	0.0712
RDens	**0.0363**
num	**0.0493**
OverAN	0.5114
OverA	0.6774
OverN	0.4418
NOverAN	0.7142
NOverA	**3.87**$*10^{-4}$
NOverN	0.1416

Table 5.3: p values on the CLC-FCE AN corpus-attested subset

We note that we can reliably distinguish between distributional vectors for correct and incorrect combinations using VLen. This confirms our original hypothesis that the vectors representing incorrect combinations are shorter than those representing correct ones. When AN vectors are derived by application of a composition function to the word distributional vectors it is assumed that the difference in distribution of the co-occurrence counts results in a shorter vector length for incorrect combinations. In the case of distributional AN vectors this phenomenon is a result of the fact that corpus-attested ANs annotated as incorrect in our data generally have lower frequency than corpus-attested ANs annotated as correct. As a result, they have lower counts along semantic space dimensions.

The measures based on calculation of the distance from the AN vector to the adjective and noun vectors do not show a difference at the given level. Possibly, corpus-attested ANs that are annotated as incorrect in learner data are not strongly semantically deviant. We also note that even though the p values for the differences in cosN and cosA on the two groups of vectors are higher than 0.05, the vectors for the correct combinations return higher values, thus are semantically more similar to the input nouns and adjectives than incorrect combinations.

Two measures based on quantitative assessment of the vectors' neighbourhood, RDens and num, show differences at the given level but return higher values for the ANs annotated as incorrect than for those annotated as correct. Neither dens nor densAll show differences between the two groups of vectors and also return higher values for the ANs annotated as incorrect. These results are similar to those obtained by Vecchi (2013) who concluded that semantically anomalous ANs might be located in artificially dense neighbourhoods surrounded by neighbours that share more in common with the input adjectives than the nouns. All four measures that are based on quantitative assessment show that vectors for the incorrect combinations are surrounded by closer neighbours than vectors for the correct combinations.

Among the measures based on qualitative assessment of the vectors' neighbourhood, only `NOverA` shows a difference at the given level, with the correct ANs being surrounded by neighbours that are more similar to the input adjectives than for the incorrect combinations. The results obtained for the other measures show that corpus-attested ANs are surrounded by neighbours that are semantically similar to or that lexically overlap with the input words, and these properties are not sufficiently different for the two groups of vectors.

Distributional AN vectors are sparser than model-generated AN vectors (see §2.4), and compositional distributional semantic models are applied to overcome the problem of data sparsity and to generate more reliable representations for the composite vectors. Not all of the proposed measures can perform well on sparser vectors. In addition, some measures may be particularly suitable for model-generated vectors as they take into account the contribution of each of the input words: for example, `cosN` and `cosA` are based on measuring how far the composite vector is moved from the input words. It is interesting to note that `VLen` for model-generated vectors is also based on the idea that the combination of vectors for two semantically incompatible words results in a composite vector with substantially different properties. However, it performs well on the distributional vectors for a different reason: since the corpus-attested ANs annotated as incorrect in the learner data are, on average, less frequent than those annotated as correct, this property manifests itself in the length of the distributional vectors. In general, we conclude that the distributional vectors for the corpus-attested ANs annotated as correct are too similar to the distributional vectors for the corpus-attested ANs annotated as incorrect to be reliably distinguished from each other with some of the proposed measures.

5.4 Experiments on the AN datasets

We present the results of our experiments on the CLC-FCE and on the controlled annotated datasets. The crucial difference between the two datasets is that the CLC-FCE dataset contains corpus-attested as well as corpus-unattested examples, and consists of adjective–noun combinations with many more input adjectives than in the controlled annotated dataset: the full CLC-FCE dataset contains combinations with $1,061$ distinct adjectives, while the annotated dataset contains only 61 input adjectives. To be able to compare the results on the datasets and check how different factors affect the results, we run a number of additional experiments on the subsets of the CLC-FCE dataset, as well as on the full CLC-FCE and annotated datasets:

- CLC-FCE dataset
  - all ANs ($4,681$ correct and 530 incorrect)
  - corpus-attested ANs ($3,294$ correct and 286 incorrect)
  - corpus-unattested ANs ($1,387$ correct and 244 incorrect)
  - ANs with the set of 61 selected most problematic adjectives ($1,185$ correct and 210 incorrect)

  - corpus-attested ANs with the selected set of adjectives ($1,029$ correct and 145 incorrect)
  - corpus-unattested ANs with the selected set of adjectives (156 correct and 65 incorrect)

- annotated dataset
  - combinations annotated OOC (630 correct and 168 incorrect ANs)
  - combinations annotated IC (406 correct and 392 incorrect ANs)

We expect certain factors to have different effect on the results, and we expect to see certain similarities in the results between the annotated dataset and the subset of corpus-unattested examples with the chosen set of adjectives from the CLC-FCE dataset. Since the CLC-FCE dataset contains error annotation for the ANs in context, we also expect the results to be closer to those on the annotated dataset with IC annotation. For the *alm* model we can only report the results on *selected* subsets of the data consisting of the ANs with the set of 61 adjectives, since for this model we generate a matrix for each adjective.

5.4.1 CLC-FCE dataset

We run our experiments on 6 different subsets starting with the full CLC-FCE AN dataset (the results are presented in Table 5.4) and then considering smaller and more specific subsets (Tables 5.5 through 5.9). We believe that the full dataset combines ANs with different properties, and setting apart specific subsets helps eliminate potentially interfering factors and assess the performance of the models on more homogeneous sets of ANs.

We use the one-shot t-test to indicate the scale of the effect observed, and for each model and semantic measure, we report the p value denoting the difference between the groups of vectors for the correct and incorrect ANs. We mark the results that support our original hypotheses at the $p < 0.05$ level in bold blue, while those that demonstrate the difference between the properties of the vectors at the given level but in the direction opposite to the original hypotheses are in bold red.

Measures of semantic anomaly

We see that `VLen` only shows a difference at the given level with the *mult* model. The *mult* model zeros or diminishes values along semantically incompatible dimensions in the composed vector whereas other models can still assign some combined value to such dimensions: for example, the *add* model aggregates the values from the two input vectors. Our tests confirm that the *mult* model generates longer vectors for the correct ANs in the *all* and *selected* datasets. However, we get results that contradict our intuition about vector length on the *all-unattested* and *selected-unattested* subsets: for the corpus-unattested examples, the *mult* model generates shorter vectors for the ANs annotated as correct than for those annotated as incorrect. We note that as the set of examples extracted from the CLC-FCE is not controlled in the same way as the examples in the annotated dataset, this result might be due to the fact that fewer examples annotated as incorrect in the CLC-FCE are truly semantically deviant.

Metric	*add*	*mult*	*alm*
VLen	0.2440	**0.0361**	–
cosN	0.6952	0.2214	–
cosA	0.8418	0.9298	–
dens	0.5649	**6.80$*10^{-5}$**	–
densAll	0.8528	**0.0010**	–
RDens	0.5047	0.5699	–
num	0.1760	0.5047	–
OverAN	0.6642	0.1845	–
OverA	0.2181	**0.0262**	–
OverN	0.5339	0.9571	–
NOverAN	0.6961	**8.82$*10^{-6}$**	–
NOverA	0.2869	**0.0027**	–
NOverN	0.2303	**8.11$*10^{-4}$**	–

Table 5.4: p values on the CLC-FCE AN dataset (all)

Metric	*add*	*mult*	*alm*
VLen	0.3365	0.1217	–
cosN	0.5739	0.2038	–
cosA	0.5462	0.2800	–
cosAN	0.9259	0.8061	–
dens	0.4290	0.3092	–
densAll	0.3072	0.4520	–
RDens	0.4488	0.2111	–
num	0.6476	0.0698	–
OverAN	0.2802	0.8222	–
OverA	0.7211	0.3704	–
OverN	0.2106	0.3387	–
NOverAN	0.7067	0.1679	–
NOverA	0.5362	0.3242	–
NOverN	0.4279	0.3510	–

Table 5.5: p values on the CLC-FCE AN dataset (all, attested)

Metric	*add*	*mult*	*alm*
VLen	0.0846	**1.70$*10^{-5}$**	–
cosN	**0.0465**	0.3351	–
cosA	0.2464	0.5717	–
dens	0.0912	**0.0162**	–
densAll	0.0596	0.0751	–
RDens	0.2179	0.4663	–
num	0.7195	0.9312	–
OverAN	0.7022	0.4217	–
OverA	**5.43$*10^{-4}$**	0.3143	–
OverN	**6.15$*10^{-4}$**	0.7115	–
NOverAN	0.9319	0.0713	–
NOverA	**0.0015**	0.5021	–
NOverN	**0.0036**	0.0756	–

Table 5.6: p values on the CLC-FCE AN dataset (all, unattested)

The two measures that are based on the relations between the input and the output vectors – `cosN` and `cosA` – show the differences at the given level on some subsets of the data with all three models. We note that the differences are more evident on the *selected* subset which contains examples for only the selected set of 61 most problematic adjectives. We also see that on these subsets the vectors modelled with the *add* model show results that contradict our original hypothesis: the vectors representing incorrect AN combinations tend to be closer to the input noun vectors than vectors representing correct ANs. At the same time, the *mult* and *alm* models generate vectors for the correct combinations that are closer to the input noun and further away from the input adjective, while the opposite holds for the vectors representing incorrect ANs. Therefore, the `cosN`

Metric	*add*	*mult*	*alm*
VLen	0.0526	**$3.25*10^{-4}$**	0.6049
cosN	**$2.25*10^{-4}$**	0.8933	0.1155
cosA	0.1870	**0.0324**	**0.0217**
dens	**0.0074**	**$1.32*10^{-6}$**	**0.0401**
densAll	**0.0267**	**$5.52*10^{-5}$**	0.1352
RDens	0.0978	0.2827	**0.0291**
num	0.0968	0.5177	**0.0116**
OverAN	0.2561	0.2596	0.1811
OverA	0.3296	**0.0020**	0.2551
OverN	0.1114	0.2376	0.4973
NOverAN	0.3012	**0.0012**	0.7618
NOverA	0.0716	**$2.62*10^{-4}$**	–
NOverN	0.2190	0.3500	0.7618

Table 5.7: p values on the CLC-FCE AN dataset (selected)

Metric	*add*	*mult*	*alm*
VLen	0.7181	0.1003	0.2942
cosN	**0.0019**	0.2735	0.1431
cosA	**0.0345**	0.2201	**0.0132**
cosAN	0.7285	0.9395	0.6572
dens	0.2406	**0.0147**	0.1218
densAll	0.4632	0.0814	0.1086
RDens	0.8930	0.4083	0.1201
num	0.9327	0.2882	**0.0099**
OverAN	0.1728	0.8487	0.1527
OverA	0.1056	**0.0467**	0.2606
OverN	**0.0326**	0.1329	0.3916
NOverAN	0.9337	0.1599	0.6049
NOverA	**0.0280**	**0.0334**	–
NOverN	**0.0275**	0.8628	0.6049

Table 5.8: p values on the CLC-FCE AN dataset (selected, attested)

Metric	*add*	*mult*	*alm*
VLen	0.6259	**0.0228**	0.9241
cosN	0.1615	0.8513	0.1759
cosA	0.3651	0.2903	0.8769
dens	0.5655	0.2909	0.6466
densAll	0.7625	0.3551	0.6138
RDens	0.6828	0.4969	0.4539
num	0.5491	0.2560	0.9186
OverAN	0.4070	0.5971	0.8539
OverA	0.8850	0.3189	0.8539
OverN	0.8025	0.3211	–
NOverAN	0.4684	0.8911	–
NOverA	0.3843	0.4096	–
NOverN	0.6464	0.5971	–

Table 5.9: p values on the CLC-FCE AN dataset (selected, unattested)

measure shows the results that we expect to obtain: the meaning of the input noun is preserved in the correct combinations and the vectors for the correct combinations are closer to the input nouns. We note that both *mult* and *alm* models generate vectors for the incorrect AN combinations that are closer to the input adjective vector than for the correct AN combinations.

The measures that are based on quantitative assessment of the vector neighbourhood in the semantic space – `dens`, `densAll`, `RDens` and `num` – with the *add* and *mult* models show the differences at the given level between the two groups of vectors that support the hypothesis of Vecchi (2013): the vectors for the ANs annotated as incorrect are placed in denser neighbourhoods than vectors for the correct combinations. This hypothesis is also supported by our tests on the distributional AN vectors presented in the previous section. It is interesting to note that with the *mult* model the measures that assess vector neighbourhood qualitatively show that even though the vectors for correct combinations are placed in sparser neighbourhoods, these neighbourhoods are populated by more similar neighbours than vectors for incorrect combinations. The *add* model places the vectors for the correct combinations in the neighbourhoods that share neighbours with the input nouns, but vectors for the incorrect combinations in the neighbourhoods that share neighbours with the input adjectives.

Model performance

All models are able to detect the difference between the generated vectors for the correct and incorrect combinations at least with some measures and on some subsets of the data. *Alm*, previously reported as a more promising model, on our data does not outperform the *add* and *mult* models. Among the three models, the *mult* model shows best results overall, since it detects the difference between the vectors more often and the results of this model are similar to those obtained on the distributional vectors (Table 5.3).

Performance on different subsets of data

Except for the occasional changes in the results for some models and measures (for example, `VLen` with the *mult* model on the *all* and *selected* subsets), performance of the models is generally consistent on the different subsets of the data. For example, the models cannot reliably distinguish between the vectors for the correct and incorrect combinations among those that are corpus-attested (Table 5.5) because many of the combinations from the CLC-FCE dataset that are annotated as incorrect may not be truly deviant or may be incorrectly used in context, so they are not sufficiently different from the combinations annotated as correct. When we specifically look at the corpus-attested combinations with the selected set of the most problematic adjectives (Table 5.8) a number of models and measures prove to be able to detect the difference. At the same time, it proved hard to distinguish between correct and incorrect combinations with the chosen set of models and measures in the corpus-unattested subset of ANs with the selected adjectives (Table 5.9).

5.4.2 Annotated dataset

As before, we report p values for each model and semantic measure. The results for the OOC annotation are reported in Table 5.10, and for the IC annotation in Table 5.11. The p values below the 0.05 level are marked in bold, with those supporting the original hypotheses in blue, and those showing the opposite results in red.

A number of measures show a difference at the given level between the two groups of vectors. For most measures that show such differences on the OOC-annotated dataset the effect is strengthened on the IC-annotated dataset: for example, `cosA` distinguishes between the *add* model-generated vectors with $p < 0.01$ on the OOC-annotated dataset, and with $p < 0.001$ on the IC-annotated dataset.

At the same time, we see some unexpected results on the IC-annotated dataset: the `VLen` measure shows that model-generated vectors for the ANs annotated as incorrect in context

Metric	*add*	*mult*	*alm*
VLen	0.7589	0.7690	0.1676
cosN	0.1621	**0.0248**	**0.0227**
cosA	**0.0029**	0.4782	0.0921
dens	0.6731	0.1182	0.1024
densAll	0.4967	0.1026	0.1176
RDens	0.2786	0.8754	0.1970
num	0.3132	0.4673	0.3765
OverAN	0.8529	0.1622	0.9663
OverA	**0.0151**	0.6377	0.5051
OverN	**0.0138**	0.0764	0.4656
NOverAN	0.6572	0.9745	0.0858
NOverA	**0.0015**	0.4436	0.1575
NOverN	**0.0018**	0.2182	0.1497

Table 5.10: p values, OOC AN annotation

Metric	*add*	*mult*	*alm*
VLen	0.6675	**0.0027**	**0.0111**
cosN	**0.0417**	**0.0070**	0.1845
cosA	$\mathbf{3.26*}10^{-5}$	0.1791	0.1442
dens	0.4756	0.7120	0.1278
densAll	0.2874	0.7139	0.6183
RDens	0.8934	0.8664	0.1985
num	0.7077	0.7415	0.4369
OverAN	0.1962	0.8635	0.6682
OverA	$\mathbf{7.20*}10^{-5}$	0.7271	0.6358
OverN	**0.0022**	0.9680	0.9867
NOverAN	0.1066	0.6304	0.1587
NOverA	$\mathbf{6.92*}10^{-6}$	0.7354	0.1576
NOverN	$\mathbf{2.89*}10^{-5}$	0.6978	0.2610

Table 5.11: p values, IC AN annotation

are longer than for the ANs annotated as correct, which contradicts our original hypothesis. However, we have seen similar results on some CLC-FCE subsets: for example, with the *mult* model on the unattested ANs (see Tables 5.6 and 5.9).

In general, the results obtained on the controlled annotated datasets follow the same patterns as the results on the CLC-FCE dataset: the *add* model performs better than other models. We note quite a consistent pattern with this model: the model-generated vectors for the correct ANs are closer to the input adjectives (`cosA`), show higher lexical overlap (`OverA`) and overlap in terms of the neighbours shared with the adjectives (`NOverA`), while the model-generated vectors for the incorrect ANs tend to stay closer to the input nouns. This is similar to what we have observed on the CLC-FCE dataset (see Table 5.8).

5.5 Experimental setting for VO combinations

For the VOs, we construct the semantic space in a similar way: we populate it with distributional vectors for a large number of *target elements* including some frequent verbs and nouns, verbs and nouns from the test combinations, as well as VOs with these verbs and nouns. Since we use a similar set of measures for detecting semantic anomaly in VOs and a number of those measures are based on exploration of composite vectors' neighbourhoods, we need additional noun and verb vectors which are not necessarily part of the test VOs, as well as some additional VO vectors to populate the semantic space.

The semantic space is represented by a matrix encoding sentence-internal word co-occurrences for the target elements and the same $10K$ context words as have been used for the ANs: this set of $10K$ context words consists of the most frequent $6,590$ nouns, $1,550$ adjectives and $1,860$ verbs in the combined corpus. The raw sentence-internal co-occurrence counts have been transformed into *LMI* scores.

We collected the target elements for the VO semantic space using a similar procedure to AN semantic space construction: we have selected the $4K$ verbs and $8K$ nouns which are most frequent in the concatenated corpus, excluding the top 50 most frequent words in each case. We also add the component verbs and nouns from the test VOs to the semantic space, and collect some frequent verbs that occur more than $100K$ but less than $740K$ times in the combined corpus. As a result, we augment the semantic space with additional noun and verb distributional vectors. The VO semantic space contains $8,357$ nouns and $4,053$ verbs.

We generate all possible VOs by crossing the set of verbs with the set of nouns, and check the generated VOs against the BNC. We randomly choose VOs that occur at least 10 times in the BNC and at least 100 times in the concatenated corpus consisting of the BNC and ukWaC. This results in a set of $42,080$ VOs. We also add to this set corpus-attested VOs from the CLC-FCE test set that are not yet included in the set of VOs. As a result, the semantic space contains $43,671$ VOs, and in total, the number of target elements equals $56,082$.

We apply SVD to the noun and verb components of the matrix, consisting of $12,410$ vectors, and then project the $43,671$ VO vectors onto the same semantic space. Using this method, we reduce the original $56,082 \times 10K$ matrix encoding VO semantic space to a $56,082 \times 300$ matrix that we use in our experiments.

5.6 Semantic models for VO combinations

We apply the same set of compositional distributional semantic models to generate the composite VO representations from the input verb and noun distributional vectors.

For the simple *add* and *mult* models, we derive the VO (vn) vectors by component-wise addition and multiplication applied to the verb (v) and noun (n) vectors:

$$vn_i = v_i + n_i \tag{5.5}$$

$$vn_i = v_i \cdot n_i \tag{5.6}$$

We also adapt the *alm* model to derive the VO combinations in a similar way to AN combinations. In §2.4, we have noted that certain words, for example, nouns, can be directly defined by their distribution, while some other words, for example, adjectives and verbs, might "adjust" their meaning depending on the words they modify. The particular algebraic representation of such words depends on their subcategorisation frame. Adjectives and intransitive verbs are first-order one-argument distributional functions which can be encoded with matrices, while transitive verbs are higher-order functions which can be encoded with *tensors* of the appropriate dimensionality depending on the number of arguments. Baroni et al. (2014a, p. 43) suggest analysing an intransitive verb as a mapping from noun space to sentence space, and representing it as a matrix of shape $K \times J$ mapping from the J-dimensional noun space in which subject vectors live onto the K-dimensional sentence space. In contrast, a transitive verb is a third-order $(K \times J) \times J$ tensor mapping from the J-dimensional noun space where object vectors live onto a VP-space, or an intransitive-verb-like $K \times J$ matrix.

The group of models that treats adjectives and verbs as distributional functions represents intransitive verbs as matrices or second-order tensors, transitive verbs as third-order tensors, and di-transitive verbs as fourth-order tensors. VOs in our datasets contain verbs with two or more arguments, and within this type of model these verbs should be represented with different structures. However, we only keep the combinations of the verbs with their direct objects. If we represent each verb with the higher-order algebraic representation, their combination with the object noun vector will result in structures of a different order: **close computer* will be represented with a matrix which then should be applied to a subject noun vector, while *tell advice* will be represented with a third-order tensor which then should be applied to a second object and a subject noun vectors.

We are primarily interested in violation of compositionality between a verb and its direct object. Therefore, for this data we do not fully model the verb argument structure with all verb arguments, and we would like to have a simple verb–object semantic representation of the same dimensionality for different types of verbs. Then, using an approach similar to that applied to the AN combinations, we can distinguish between semantic representations for the acceptable and deviant VOs.

Baroni et al. (2014a) show how tensor representations for verbs can be learned from the observed examples using a multi-step regression algorithm. The procedure is similar to that used to learn matrices for the adjectives in *alm* models. Since we address the verb–object pairs only, we believe that the verbs can be encoded as matrices with one argument slot for the direct object encoded with distributional noun vectors. The matrices can be learned from the observed examples of VOs with the particular verb using one-step regression. The matrices then will encode the interaction of the semantic components of the verb and its direct objects in the observed VOs. When applied to the input noun vectors, they will map from the noun space to the space of VOs, where the derived VOs are encoded with model-generated vectors. This representation is different from that commonly used in compositional distributional semantics (Clark, 2015; Baroni et al., 2014a; Grefenstette et al., 2013), however, we think that for this task a matrix-based representation of the verbs is justified.

We refer to this model as *verb-specific linear maps* (*vlm*), and for each verb from our test set we estimate the weight coefficients with multivariate partial least squares regression using the R `pls` package (Mevik and Wehrens, 2007) and applying the leave-one-out training regime. Similarly to the adjective case, we change the number of latent variables used by the training algorithm depending on the number of available noun–VO training pairs. In our experiments, this number ranges from 3 to 40. As a result, a weight matrix encoding verb–object interaction for each verb in the test set is obtained.

We apply the three models of semantic composition to generate VO vectors, and use the same procedure to distinguish between vectors representing correct VOs and incorrect VOs (see §5.2). We use one-shot unpaired t-tests assuming a two-tailed distribution, and report the difference to indicate the scale of the effect observed. The p values lower than 0.05 are marked in bold.

5.7 Measures of semantic anomaly for VO combinations

Below, we list the measures that we apply to the model-generated VO vectors. These measures are based on those applied to the ANs. As before, we group the 14 measures by their type based on the underlying hypotheses:

1. **Measures based on the properties of the model-generated vector**:
   - **Vector length (*VLen*)** – see §2.4 for a description.
2. **Measures based on the relations between the input and the output vectors**:
   - **Cosine to the input noun (*cosN*)** – see §2.4 for a description.
   - **Cosine to the input verb (*cosV*)** is analogous to `cosA` proposed for the ANs: we hypothesise that the model-generated VO vector should be situated further away from the vector for the input verb since in anomalous VOs the original verb meaning is distorted. We can also expect a stronger effect for the `cosV` measure than for `cosN` measure since VO combinations are verb-like rather than nominal-like.
   - **Cosine to the distributional vector for the corpus-attested VOs (*cosVN*).** For the VOs which are attested in the corpus, we measure the distance from the model-generated vector to the distributional vector and assume that semantically acceptable VOs should be located closer to their distributional vectors than semantically deviant and non-compositional ones.
3. **Measures based on the neighbourhood of the model-generated vectors**:
   - **Density of the neighbourhood populated by 10 nearest neighbours (*dens*)** – see §2.4 for a description.
   - **Density among the 10 nearest neighbours (*densAll*)** is estimated as an average of the 11 density values calculated for each member of the set {*VO vector*, *its* 10 *nearest neighbours*}, where each density value is estimated as an average distance from the member of the set to all other members of the set.
   - **Ranked density in close proximity (*Rdens*)** is estimated as a weighted sum of the distances of the neighbours within *close proximity*, where each neighbour's distance value is weighted with its rank. It is assumed that semantically acceptable and correct VOs have more close neighbours and, as a result, get higher values for the `RDens` measure.
   - **Number of neighbours within close proximity (*num*)** is used as a separate measure and is assumed to have lower values for incorrect combinations.
   - **Lexical overlap between the 10 nearest neighbours and constituent noun and verb (*OverVN*)** is estimated as the proportion of the 10 nearest neighbours containing the same constituent words as in the tested VOs. We

assume that correct VOs are surrounded by words and combinations similar to the input noun and verb: for example, we might expect to see the verb *close* and noun *door* among the 10 nearest neighbours of *close door*, as well as some other VOs with *close* and *door*. At the same time, we assume that **close computer* is less semantically similar to either *close* or *computer*, and we expect to see lower number of words and VOs with *close* or *computer*.

- **Lexical overlap between the 10 nearest neighbours and input noun (*OverN*)** is a variant of `OverVN` for the input noun.
- **Lexical overlap between the 10 nearest neighbours and input verb (*OverV*)** is a variant of `OverVN` for the input verb.
- **Overlap between the 20 nearest neighbours for the VO and the nearest neighbours for the constituent noun and verb (*NOverVN*)** is estimated as the proportion of the common neighbours among the 20 nearest neighbours for the model-generated VO and the 20 nearest neighbours for the distributional vectors of the constituent words, and we assume that correct VOs and their constituent words should be placed in similar neighbourhoods.
- **Overlap between the 20 nearest neighbours for the VO and the nearest neighbours for the constituent noun (*NOverN*)** is a variant of `NOverVN` for the input noun.
- **Overlap between the 20 nearest neighbours for the VO and the nearest neighbours for the constituent verb (*NOverV*)** is a variant of `NOverVN` for the input verb.

5.7.1 Tests on corpus-attested word combinations

The CLC-FCE dataset contains $4,557$ corpus-attested VO combinations, $3,997$ of which are annotated in the learner data as correct and 560 are annotated as incorrect. We run the first set of experiments on the distributional VO vectors. The results are reported in Table 5.12, and as before we mark the metrics that show differences at the given level in bold, the results that support our original hypotheses in blue, and those that show differences in the opposite direction in red.

We note again that corpus-attested combinations that are annotated as correct in the learner data, in general, have higher frequency than those that are annotated as incorrect. We assume that this can be used as a reliable cue for detecting incorrectness in corpus-attested combinations. When the vectors for the VOs are built from the distributional data, lower frequency of the incorrect combinations results in lower co-occurrence counts along vector dimensions, and, as a consequence, in lower vector length for deviant VOs. Similar to the ANs, the `VLen` measure performs well on corpus-attested combinations and distinguishes between the vectors for correct and incorrect combinations most reliably.

`CosN` and `cosV` show differences at the given level between the distributional vectors for correct and incorrect VOs, but return higher values for the incorrect VOs. That shows that, contrary to our prediction, the distributional vectors for the incorrect VOs stay closer to the verb and noun vectors. Not all of the VOs annotated as incorrect in the learner

Metric	*distributional*
VLen	$\mathbf{1.79}*10^{-22}$
cosN	**0.0434**
cosV	$\mathbf{5.52}*10^{-4}$
cosVN	1
dens	**0.0235**
densAll	0.0538
RDens	0.4837
num	0.3735
OverVN	0.4978
OverV	0.5116
OverN	0.1508
NOverVN	0.5361
NOverV	0.5496
NOverN	0.8668

Table 5.12: p values on the CLC-FCE VO corpus-attested subset

data show strong semantic deviance and some of them are annotated as incorrect due to their mismatch with the surrounding context: for example, **meet world* corrected to *enter world* and **learn knowledge* corrected to *acquire knowledge* are clearly more deviant than some other examples annotated as incorrect including $^{(*)}$*miss course* corrected to *miss class* or $^{(*)}$*understand question* corrected to *understand request*. The verb type might also have an effect. For example, the correct combinations with 'light' verbs can be situated further away from the input verbs: *do shopping*, while being correct, has less to do with the verb *do* than with the verb *shop*, while *make plan* can be expected to be closer to *plan* than to *make*. We assume that such factors may have a certain effect, but we still expect to obtain the predicted results when the vectors are generated with the models rather than extracted from the data.

The `dens` measure shows a difference at the given level and supports our original hypothesis: the distributional vectors for VOs annotated as incorrect in the learner data are located in sparser neighbourhoods in the semantic space than those annotated as correct. We also note that the `densAll` measure shows results close to significance threshold and returns higher values for the correct VOs.

The other measures do not show differences at the given level of $p < 0.05$, and we conclude it might be due to reasons similar to those outlined for the corpus-observed ANs (see §5.3).

5.8 Results on VO combinations

We run the experiments on CLC-FCE and on the controlled annotated dataset, setting apart different subsets of the data. The full CLC-FCE VO dataset contains combinations with 603 distinct verbs, while the annotated dataset contains only 77 input verbs.

We run the experiments on the following subsets in our data:

- CLC-FCE dataset
  - all VOs (4, 911 correct and 789 incorrect)
  - corpus-attested VOs (3, 997 correct and 560 incorrect)
  - corpus-unattested VOs (914 correct and 229 incorrect)
  - VOs with the set of 77 selected most problematic verbs (1, 378 correct and 316 incorrect)
  - corpus-attested VOs with the selected set of verbs (1, 166 correct and 224 incorrect)
  - corpus-unattested VOs with the selected set of verbs (212 correct and 92 incorrect)
- annotated dataset
  - combinations annotated OOC (445 correct and 356 incorrect)
  - combinations annotated IC (314 correct and 487 incorrect)

5.8.1 CLC-FCE dataset

We run the experiments on 6 subsets of the data starting with the full CLC-FCE VO dataset (the results are presented in Table 5.14) and then considering smaller and more specific subsets (Tables 5.14 through 5.18). For each model and semantic measure, we report the p value, marking the values lower than 0.05 in bold.

Measures of semantic anomaly

VLen shows differences at the given level in a number of subsets. However, in all cases the model-generated vectors for the combinations annotated as incorrect are longer than the model-generated vectors for the combinations considered to be correct. Although this contradicts our original hypothesis, this result is similar to those obtained on some of the AN datasets.

Both cosV and cosN show differences at the given level of $p < 0.05$ with all models on at least some subsets of VOs. In all cases, the model-generated vectors for the correct combinations are closer in the semantic space and, thus, semantically more similar to both input noun and input verb. We also note that the cosV measure shows differences at the given level on more subsets than cosN. CosN mostly works in combination with the *add* model, while cosV shows differences on the vectors modelled by the *mult* and *vlm* models.

The group of measures based on the quantitative assessment of the vectors' neighbourhood – dens, densAll, RDens and num – shows differences at the given level of $p < 0.05$ with all the models at least on some subsets. In all cases the model-generated vectors for the incorrect VOs are located in denser neighbourhoods and are surrounded by higher number of close neighbours than vectors for the correct combinations. Although this result contradicts the original hypotheses, it follows the same pattern as we observed for the ANs.

Metric	*add*	*mult*	*vlm*
VLen	**$1.01*10^{-4}$**	0.0968	–
cosN	**$6.89*10^{-5}$**	0.2522	–
cosV	0.1403	0.0522	–
dens	**$7.59*10^{-4}$**	**$2.31*10^{-6}$**	–
densAll	**0.0442**	**0.0010**	–
RDens	**$5.89*10^{-4}$**	**$1.61*10^{-3}$**	–
num	**0.0040**	0.1860	–
OverVN	0.4177	**$7.80*10^{-5}$**	–
OverV	**0.0297**	**0.0047**	–
OverN	**0.0015**	**0.0018**	–
NOverVN	**0.0015**	**0.0024**	–
NOverV	**$1.13*10^{-4}$**	0.1352	–
NOverN	**0.0117**	**$5.85*10^{-4}$**	–

Table 5.13: p values on the CLC-FCE VO dataset (all)

Metric	*add*	*mult*	*vlm*
VLen	**0.0028**	0.1340	–
cosN	**$1.09*10^{-4}$**	0.2235	–
cosV	0.3687	**0.0233**	–
cosVN	**$4.38*10^{-4}$**	**$1.35*10^{-6}$**	–
dens	0.0737	**$1.80*10^{-7}$**	–
densAll	0.3442	**$7.95*10^{-5}$**	–
RDens	**0.0055**	**0.0027**	–
num	**0.0152**	0.3471	–
OverVN	0.9871	**$5.65*10^{-5}$**	–
OverV	0.2350	**$8.28*10^{-4}$**	–
OverN	**0.0228**	**0.0128**	–
NOverVN	**$1.76*10^{-4}$**	**0.0016**	–
NOverV	**0.0016**	0.0900	–
NOverN	0.1172	**$5.27*10^{-4}$**	–

Table 5.14: p values on the CLC-FCE VO dataset (all, attested)

Metric	*add*	*mult*	*vlm*
VLen	0.6278	**0.0047**	–
cosN	0.8704	0.5256	–
cosV	0.6344	0.3420	–
dens	0.5373	0.0995	–
densAll	0.8035	0.2386	–
RDens	0.6833	0.2236	–
num	0.1945	0.3897	–
OverVN	0.2948	0.6214	–
OverV	0.3376	0.5673	–
OverN	0.9960	0.9956	–
NOverVN	0.3587	0.8897	–
NOverV	0.6369	0.8274	–
NOverN	0.9514	0.9479	–

Table 5.15: p values on the CLC-FCE VO dataset (all, unattested)

Even though the model-generated vectors tend to have denser neighbourhoods, qualitative analysis shows that these neighbourhoods are populated by lexically and semantically less similar words and phrases: we observe the differences at the given level for the measures `OverVN` through to `NOverN`, in particular, for the vectors generated with the *mult* model. *Vlm* does not show differences with this group of measures, while the results for the *add* model reveal a consistent pattern: the *add* model-generated vectors for the correct VOs tend to have more in common with the input noun, while the model-generated vectors for the incorrect combinations share more in common with the input verb. Semantic similarity of the model-generated VO vectors with the input noun for the *add* model is supported by the results for the `cosN` measure.

Metric	*add*	*mult*	*vlm*
VLen	**$6.29*10^{-4}$**	0.2281	0.6766
cosN	**0.0238**	0.8670	0.3239
cosV	0.9574	**$5.45*10^{-5}$**	**0.0495**
dens	0.2143	**$2.21*10^{-4}$**	0.3486
densAll	0.3261	**$1.13*10^{-5}$**	0.3141
RDens	0.1059	**0.0019**	0.1921
num	**0.0377**	0.0600	0.6258
OverVN	0.3261	**$1.13*10^{-5}$**	0.3141
OverV	**0.0246**	**$6.94*10^{-4}$**	0.1334
OverN	0.1747	**0.0430**	0.5812
NOverVN	0.0781	**$4.25*10^{-5}$**	0.8580
NOverV	0.1778	**$7.23*10^{-4}$**	0.5618
NOverN	0.5928	**0.0175**	0.7780

Table 5.16: p values on the CLC-FCE VO dataset (selected)

Metric	*add*	*mult*	*vlm*
VLen	**$3.54*10^{-5}$**	0.2570	0.8576
cosN	**$7.26*10^{-5}$**	0.8190	0.8510
cosV	0.0873	**$4.09*10^{-5}$**	**0.0028**
cosVN	**0.0016**	**$1.33*10^{-4}$**	0.0516
dens	0.6847	**0.0030**	0.4202
densAll	0.7312	**0.0020**	0.5829
RDens	0.1075	0.1788	0.0829
num	**0.0422**	0.9293	0.4924
OverVN	0.0727	**$5.44*10^{-4}$**	0.3029
OverV	**$5.41*10^{-4}$**	**0.0065**	0.1917
OverN	**0.0108**	**0.0260**	0.8932
NOverVN	0.1057	**$1.19*10^{-4}$**	0.6015
NOverV	**0.0049**	**0.0075**	0.6273
NOverN	**0.0186**	**$2.92*10^{-4}$**	0.6296

Table 5.17: p values on the CLC-FCE VO dataset (selected, attested)

Metric	*add*	*mult*	*vlm*
VLen	0.0549	0.3356	0.2975
cosN	0.0585	0.7447	0.1816
cosV	**0.0018**	0.4063	0.4533
dens	0.9823	0.0613	0.3988
densAll	0.4508	0.1200	0.4476
RDens	0.2469	0.1171	0.9619
num	0.6494	0.1231	0.9927
OverVN	0.2853	0.0626	0.2727
OverV	0.1614	0.0537	0.8735
OverN	**0.0094**	0.7302	0.2720
NOverVN	0.2783	0.4764	0.3091
NOverV	**0.0253**	0.2704	0.3202
NOverN	**0.0016**	0.7909	0.4569

Table 5.18: p values on the CLC-FCE VO dataset (selected, unattested)

Model performance

We conclude that, similarly to the ANs, the *mult* model performs better overall than the *add* and the *vlm* models. The *vlm* model, in spite of its expected superior performance at modelling asymmetric relations, reliably distinguishes between the vectors for the correct and incorrect combinations only with a limited subset of the proposed measures. The *add* model shows differences at the level of $p < 0.05$ with a number of measures, but for some of those it returns results contrary to our expectations.

Performance on different subsets of data

We note that we get very consistent results on the general datasets and the subsets of attested combinations (see Table 5.13 and 5.14, Table 5.16 and 5.17). Discriminating between vectors for correct and incorrect VOs unattested in the corpus proved to be harder

Metric	*add*	*mult*	*vlm*
VLen	0.2871	**0.0020**	**0.0423**
cosN	0.3411	0.6064	0.6409
cosV	**$2.49*10^{-8}$**	**$9.28*10^{-8}$**	0.9319
dens	**$4.84*10^{-5}$**	**0.0189**	**$3.61*10^{-4}$**
densAll	**$6.84*10^{-6}$**	**0.0471**	**0.0101**
RDens	**$1.61*10^{-10}$**	0.0878	**0.0067**
num	**$5.04*10^{-7}$**	**0.0031**	0.7805
OverVN	0.3221	**0.0462**	0.3458
OverV	**$3.78*10^{-7}$**	**0.0137**	0.7586
OverN	**$8.04*10^{-10}$**	0.9267	0.3055
NOverVN	0.1310	0.1307	0.1147
NOverV	**$3.23*10^{-6}$**	**0.0170**	0.1377
NOverN	**$2.87*10^{-10}$**	0.7815	0.5042

Table 5.19: p values, OOC VO annotation

Metric	*add*	*mult*	*vlm*
VLen	0.1023	**$3.84*10^{-8}$**	**0.0086**
cosN	0.0831	0.2317	0.4370
cosV	**$1.78*10^{-13}$**	**$3.58*10^{-5}$**	0.2559
dens	**$3.19*10^{-7}$**	**0.0020**	**0.0364**
densAll	**$1.62*10^{-7}$**	**0.0049**	0.1971
RDens	**$5.05*10^{-9}$**	**0.0033**	0.1378
num	**$1.22*10^{-5}$**	**$3.15*10^{-4}$**	0.9765
OverVN	0.0716	**0.0446**	0.6164
OverV	**$9.00*10^{-9}$**	**0.0275**	0.9265
OverN	**$2.41*10^{-13}$**	0.7487	0.2079
NOverVN	0.2378	0.0696	0.1839
NOverV	**$2.10*10^{-8}$**	**0.0411**	0.1376
NOverN	**$4.68*10^{-15}$**	0.6638	0.7543

Table 5.20: p values, IC VO annotation

for the chosen models and using the proposed measures: on the unattested subset of *all* VOs only the *mult* model shows differences at the given level with the `VLen` measure, and this result contradicts our prediction (Table 5.15), while on the unattested subset of the VOs with the *selected* set of verbs only, the *add* model detects the difference between the two groups of vectors with a smaller number of measures than on the attested examples.

We hypothesise that the difficulty in distinguishing between the unattested VOs that are annotated as correct and those annotated as incorrect in the learner data can result from the fact that these examples are extracted in a less controlled way. The two groups of vectors prove to be similar to each other with respect to a number of their properties.

5.8.2 Annotated dataset

We report the results obtained on the OOC-annotated dataset in Table 5.19, and those for the IC-annotated dataset in Table 5.20, marking the p values lower than 0.05 in bold.

We note that all but two measures (`cosN` and `NOverVN`) detect differences at the given level between the two groups of vectors with at least one of the models. This result supports our original assumption that `cosV` should be expected to detect the difference between the vectors more reliably than `cosN`.

We observe the same pattern as for the ANs: the measures that show differences at the level of $p < 0.05$ on the OOC-annotated dataset also show them on the IC-annotated dataset. The only exception is the *vlm* model, but this model also detects the difference with a lower number of measures in general.

The results obtained on the annotated dataset follow the same patterns as the results on the CLC-FCE dataset. Both *add* and *mult* models perform well on the annotated dataset. In particular, the *add* model demonstrates consistent results on the controlled dataset and the CLC-FCE unattested VO subset with the *selected* set of verbs (cf. Table 5.18).

5.9 Discussion

To summarise, we have applied three models of compositional distributional semantics to the AN and VO combinations. In spite of their theoretical weaknesses, the simple *add* and *mult* models of Mitchell and Lapata (2008) showed promising results in our experiments. From the practical point of view, these models do not require any further training and are easy to implement. The *alm* model and its adaptation to the VO combinations that we have used in this work, in spite of its theoretical strengths and some promising results in Baroni and Zamparelli (2010) and Vecchi et al. (2011), did not outperform the simpler models on this task. We conclude that, overall, the *mult* model performs the best of the three models in our experiments.

In this work, we have proposed and applied a number of measures for detecting errors in content word combinations. These measures help detect the peculiar properties of the model-generated vectors, their relation to the input words and properties of their neighbours in the semantic space. We conclude that, overall, the measures based on assessment of the neighbourhood of the model-generated vectors consistently show good results.

The measures based on quantitative assessment of the semantic space neighbourhood of the model-generated vectors showed counter-intuitive results for most of the models and on most of the datasets. We have originally hypothesised that vectors for the correct combinations are located in denser neighbourhoods which should manifest itself in higher density (`dens` and `densAll` measures) and higher number of close neighbours (`RDens` and `num` measures). However, the results consistently show that the vectors for the correct combinations are placed in sparser neighbourhoods than vectors for the incorrect combinations. At the same time, our results support those obtained by Vecchi (2013), who has shown that the close neighbours for unacceptable ANs are more often similar to the meaning of the component adjective than the close neighbours for acceptable ANs which are more often similar to the meaning of the component noun. Vecchi (2013) concludes that in anomalous ANs the meaning of the AN is "pulled" by the adjective further away from the original noun meaning, possibly to the artificially dense area populated by adjective-related neighbours. We note that although the measures based on quantitative assessment of the neighbourhood of the model-generated vectors yield results that contradict the original hypothesis, the measures aimed at assessing whether the neighbours are semantically similar to the content word combinations show more consistent results. Our experiments show that even though the vectors for the correct content word combinations are placed in sparser neighbourhoods, they are surrounded by neighbours which are semantically more similar to them than the vectors for the incorrect combinations.

In conclusion, our experiments show that compositional distributional semantic models can reliably distinguish between the groups of vectors representing correct and incorrect content word combinations with a number of proposed measures. Therefore, the values of these semantic measures can be used to derive discriminative features for an ML classifier.

Chapter 6

Error detection algorithm

We have implemented a supervised classifier which uses output of the semantic models described in Chapter 5 as features. We describe the general principles of implementing an ED algorithm as an ML classifier in §6.1, while the practical implementation issues are discussed in §6.2. The results are presented in §6.3. We discuss the performance of the algorithm and analyse the most difficult cases for ED in §6.4, and conclude with §6.5.

6.1 Error detection as classification

In §2.3.1, we discussed the general approach to ED and EDC in function words. We noted that since function words belong to closed word classes, the set of possible confusions and, therefore, corrections is a finite set in which the number of elements can be determined in advance. This makes ML classifiers particularly suitable for performing ED on function words. We have also noted that casting ED in content word combinations as a multi-class classification task similar to that used on function words is much harder.

For instance, we can train a classifier on the multiple correct uses of a function word using the surrounding words as some of the informative features. At the same time, if we wanted to treat EDC in the use of adjectives in ANs in a similar way, we would run into a problem of data sparsity even for the most frequent adjectives and most typical of their uses. The use of adjectives cannot generally be described in terms of multi-class classification with a limited number of classes. However, since we distinguish between error detection and error correction and argue that ED is an important component of the system even if the correction is not presented, we cast it as a binary classification task of identifying for each content word combination whether it is correct or incorrect.

Selection of relevant and informative features is an important issue in ML. The features should be chosen so that they can reliably distinguish between the correct and incorrect cases, while being recurrent and generalisable enough. We noted earlier (see §2.3.1) that the informative features for function words can be extracted from their surrounding contexts. Two aspects facilitate using contexts for ED in function words: they are strong predictors of the function words, and they are highly recurrent. At the same time, the surrounding contexts cannot be as easily used to generate features for ED in content words. For example, extracting the surrounding words from a context such as *I like listening to *classic music* would not help identify an error in this particular sentence or in *I have tried a *classic dance already.*

Instead, we use the output of the compositional distributional semantic models presented in Chapter 5, where each measure is used as a separate feature for an ML classifier. Such features can be expected to be informative since our experiments confirm that they can detect the difference in the correct and incorrect combinations, while also being recurrent since every combination can generate the same set of features.

6.2 Implementation of the algorithm

6.2.1 Theoretical background

The choice of a particular ML classifier for the task should depend on the problem addressed, the hypothesis space and the way the features describe the instances to be classified. We have run preliminary tests with some classifier models, including Support Vector Machines and Decision Trees, and the best results so far were obtained with the *Decision Tree* (*DT*) classifier.

DT is a conceptually simple algorithm with high expressive power. Its clear advantages are that it is easy to implement and understand while being sufficiently effective. It has previously been successfully applied to a number of tasks including grammatical EDC (Gamon, 2010). In essence, decision trees encode logical expressions of the form *if ... then ...* and are able to combine multiple conditions. The different conditions which are based on features in the feature space can interact with each other at the different nodes within the tree.

Decision trees correspond to DNF ($\vee$) expressions: every tree can be described with a logical expression in terms of DNF traversing the tree from its root via every branch to the leaves. It follows from this, that decision trees are maximally expressive: the only data that they cannot separate is data that is inconsistently labelled, i.e., when the same instance appears twice with different labels (Flach, 2012).

Each internal node in a tree is labelled with a feature, and each edge emanating from an internal node is labelled with a literal. The set of literals at a node is called a *split*. Each leaf of the tree represents a logical expression, which is the conjunction of literals encountered on the path from the root of the tree to that leaf. The extension of that conjunction (the set of instances covered by it) is called the *instance space segment* associated with the leaf (Flach, 2012, p. 132).

A tree is a compact way of representing a number of conjunctive concepts in the hypothesis space. The tree induction learner tries to identify homogeneous groups within the instance space, that would be homogeneous enough to be labelled with a single label. *Labels* in the tree that describe a binary problem, like the one that we address here, are simply *+1/-1* or *correct/incorrect*. The *literals* are the features in the feature space to be put at the root of the tree or at the splits.

For our classification task, we assume that the set of instances D_i is homogeneous, so that the function $Label(D_i)$ can return the class label for all the instances within D_i. Since we address a binary task, we can denote the set of instances as $D^{\oplus}$ for the correct and $D^{\ominus}$ for the incorrect combinations. Let's assume that we use Boolean features, so that D is split into D_1 and D_2 (or, similarly, $D_1^{\oplus}$ and $D_2^{\oplus}$, and so on). The question is how to assess the utility of a feature in terms of separating the examples into positives and negatives. The best case scenario is when $D_1^{\oplus} = D^{\oplus}$ and $D_1^{\ominus} = \varnothing$, or when $D_1^{\oplus} = \varnothing$ and $D_1^{\ominus} = D^{\ominus}$, in other words, when the feature unambiguously defines the instances of one class. In that case, the children of the split are said to be *pure*.

In practice, this will rarely be the case. Therefore, for each split we need to measure the impurity of a set of $n^{\oplus}$ positives and $n^{\ominus}$ negatives. The impurity can be defined in terms of the proportion $\dot{p} = n^{\oplus}/(n^{\oplus} + n^{\ominus})$ which also estimates the probability of the positive class. Impurity should only depend on the relative magnitude of $n^{\oplus}$ and $n^{\ominus}$, and should not change if we swap the positive and negative class (or replace $\dot{p}$ with $(1 - \dot{p})$). We also want the function defining impurity to be 0 whenever $\dot{p} = 0$ or $\dot{p} = 1$, and reach its maximum for $\dot{p} = 1/2$. One of the functions that meets these conditions and is frequently used in practice is *entropy* which is estimated as:

$$-\,\dot{p}\ log_2\dot{p} - (1 - \dot{p})\ log_2(1 - \dot{p}) \tag{6.1}$$

Entropy measures, in bits, the amount of information that is conveyed by somebody telling us the class of a randomly drawn sample. For example, the class of a pure set of instances is predictable, therefore the amount of new information is very low, and so is the entropy; for less homogeneous sets of instances where the class is less easily predictable the entropy is higher. Entropy is one of the impurity measures that is widely used in decision trees, and was introduced by Quinlan (1986).

If we denote the impurity of a single leaf D_i in the tree as $Imp(D_i)$, then the impurity of a set of mutually exclusive leaves $\{D_1, ..., D_n\}$ can be defined as a weighted average:

$$Imp(\{D_1, ..., D_n\}) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} Imp(D_i) \tag{6.2}$$

where $D = D_1 \cup ... \cup D_n$.

Using impurity estimation, we can assess the ability of each feature to split a parent node D into leaves $D_1, ..., D_n$. For that, we look at the purity gain $Imp(D) - Imp(\{D_1, ..., D_n\})$. Since we use entropy to measure purity of the splits, the purity gain is called the *information gain* splitting criterion as it measures the increase in information about the class gained by including the feature, so that the feature which contributes to the highest information gain is chosen at each step. If we are looking for the best split of the instances with the same parent, the impurity of the parent itself can be ignored and we can look for the feature which results in the lowest weighted average impurity of the children.

Flach (2012, p. 137) defines the algorithm described above as follows:

```
Data: data $D$; set of features $F$.
Result: feature $f$ to split on.
$I_{min} \leftarrow 1$;
for each $f \in F$ do
    split $D$ into subsets $D_1, ..., D_n$ according to the values $v_i$ of $f$;
    if $Imp(\{D_1, ..., D_n\}) < I_{min}$ then
        $I_{min} \leftarrow Imp(\{D_1, ..., D_n\})$;
        $f_{best} \leftarrow f$;
    end
end
return $f_{best}$
```

Algorithm 1: `BestSplit-Class(`D, F`)` – find the best split for a decision tree

The learning process goes as follows: given the training data with the set of features, the algorithm learns how to partition the training instance space given the set of features and their values. It grows the tree for the training data, identifying the order of feature application on the basis of the purity of splits at each point. Given the test data, the algorithm is able to assign the classes learned on the training data instances to the unlabelled test instances. Thus, the tree generalises the training data.

6.2.2 Practical implementation

We have used the DT classifier implementation provided with the *NLTK* toolkit (Bird et al., 2009). Additionally, we apply some modification to the feature space.

The DT classifier puts a particular feature value at each node: for example, if we could describe our classification task with binary-valued features, the values that the classifier would consider at each split would be in the form *if feature*$_i$ $= True$ *then ... else ...*. However, since the semantic measures return real numbers, our feature set consists of real-valued features. In general, it is possible to use real-valued features with a decision tree classifier: the logical expressions used in that case might look like *if* `cosN` $= 0.6891$ *then ...*, but it is clear that in our case such a tree would grow unmanageably large. At the same time, we might not want to distinguish between close feature values, for example, between `cosN` $= 0.6891$ and `cosN` $= 0.6892$ or even `cosN` $= 0.7000$. The first step is to find a way to optimise the feature space to make it easier for the feature selection algorithm to order the features.

We apply feature binning which allows us to convert each real-valued feature in our original feature set into a set of binary features for each bin. For example, suppose we have a range of values for the `cosN` feature which are spread in the range of $[-1, +1]$. We can split this range in 20 uniform bins each spanning 0.10 of the range, so that we have the set of bins $\{bin_1 = [-1.0, -0.9)$, ..., $bin_{20} = [+0.9, +1.0]\}$. The classifier will then go through each of the bins checking whether the `cosN` value falls within a particular range. For example, the classifier can learn rules like "*if '*`cosN` $\in bin_1$*' is* `True` *then return class* -1" which is equivalent to learning that if $-1.0 \leq$ `cosN` < -0.9 the combination should be assigned to the class of incorrect instances. In practice, however, setting the bins in a uniform manner is likely to produce a set of uninformative features. For example, the cosN values in our dataset rarely fall below 0 so the classifier will not learn much from the first 10 bins if those are set up uniformly. At the same time, the values are more densely distributed between 0.2 and 1.0 with some of the ranges containing more values than others. This suggests that some important distinctions can be missed by the ML algorithm if the bins always cover a predefined range that does not reflect the actual distribution of the values.

Instead, we look at the distribution of the values and set the bins accordingly. Table 6.1 shows how values for the cosN feature are distributed in our data for the ANs and VOs, and how bins are formed according to the distribution of values.

Bin	$\texttt{cosN}_{ANs}$	$\texttt{cosN}_{VOs}$
1	$[0.178, 0.458)$	$[0.103, 0.238)$
2	$[0.458, 0.809)$	$[0.238, 0.443)$
3	$[0.809, 0.870)$	$[0.443, 0.653)$
4	$[0.870, 0.892)$	$[0.653, 0.756)$
5	$[0.892, 0.906)$	$[0.756, 0.818)$
6	$[0.906, 0.917)$	$[0.818, 0.853)$
7	$[0.917, 0.928)$	$[0.853, 0.887)$
8	$[0.928, 0.936)$	$[0.887, 0.912)$
9	$[0.936, 0.948)$	$[0.912, 0.939)$
10	$[0.948, 0.969]$	$[0.939, 0.979]$

Table 6.1: Distribution of cosN values in the AN and VO combinations

The general idea behind this procedure is that each bin now contains approximately the same number of feature values, and this allows us to group feature values that are close to each other while also attaining a suitable level of feature granularity and reducing feature space dimensionality. The classifier then learns the order of feature application using *information gain*: for example, it might learn that checking whether a cosN value falls within bin_{10} reliably identifies instances of class +1 and place this rule closer to the root of the DT.

Most of the features have their values ranging from 0 (or, theoretically possible -1 for the cosine measures) to +1. VLen, RDens and num have their values in a different range, so we apply normalisation by dividing the values by the maximum value for each feature. This normalisation step simply puts the values for the three features on the same scale as the values of other features.

In all our experiments on the annotated dataset, we apply 5-fold cross-validation and report average accuracy over the folds. The full set of ANs and VOs is split into 5 subsets with 80% in each of the splits used for training and 20% for testing. We keep the AN and VO error rate in the training and test sets, as well as for each adjective or verb, approximately the same across the splits to avoid any bias.

The full feature set contains 14 features, with 13 features derived from the semantic measures, and 1 feature which represents adjective or verb identity. We hypothesise that the introduction of this feature might help the classifier learn that, for example, an AN containing *classic* has a higher chance of being incorrect, as most of the ANs with this adjective in the learner data are incorrect and involve confusion with *classical*. We also hypothesise that it facilitates learning correlation between the adjective and other feature values: it might be the case that ANs with an adjective adj_1 have on average higher `cosN` values than ANs with an adjective adj_2. This feature helps the classifier establish such dependencies between the adjective and the values of the semantic measures. For instance, in our data, ANs with the adjective *true* have significantly higher cosines between AN vectors and vectors for their constituent nouns than ANs with the adjective *false*. Intuitively, the ANs with *true* are closer to the original meaning of the noun: for example, *true happiness* is more similar to *happiness* than *false happiness* is.

6.3 Results

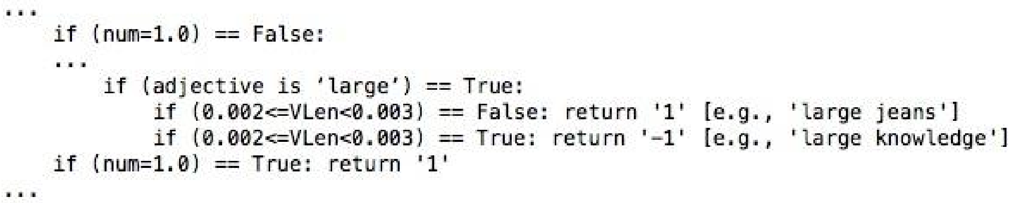
In this section, we present the results of the ED experiments on the annotated AN and VO datasets.

6.3.1 AN dataset

The best results in these experiments were obtained with the *mult* model. We note that in the experiments with the semantic models in Chapter 5 the *mult* model generally outperformed other models as well.

We ran ablation tests incrementally removing features that did not improve classifier performance in order to find an optimal feature set. The best-performing feature set for the *mult* model on the OOC annotation uses a combination of three features including *adjective*, `cosN` and `RDens`, while for the IC annotation the best-performing feature set includes *adjective*, `VLen`, `densAll`, `RDens`, `num`, `NOverA` and `NOverN` features. In general, the measures that assess the semantic neighbourhood of the model-generated vectors, such as the ones based on density or neighbours overlap, have performed well in the experiments described in Chapter 5 as well (see Tables 5.5 to 5.9).

At the same time, the sets of best performing features in the classification experiments do not exactly coincide with the semantic measures that showed the highest differences on the annotated AN dataset (Tables 5.10 and 5.11). We conclude that although the p values reported in Tables 5.10 and 5.11 show that some semantic measures can distinguish one group of ANs from another, when the measures are used as features for a classifier the results depend on how these features interact with each other as well as on their



```
...
    if (num=1.0) == False:
    ...
        if (adjective is 'large') == True:
            if (0.002<=VLen<0.003) == False: return '1' [e.g., 'large jeans']
            if (0.002<=VLen<0.003) == True: return '-1' [e.g., 'large knowledge']
    if (num=1.0) == True: return '1'
...
```

Figure 6.1: *Decision Tree* classifier pseudocode.

individual discriminativeness across the testset. We have also found that the feature encoding *adjective* interacts with other features in the feature set. Figure 6.1 illustrates a small part of the DT constructed using the best performing feature set on the IC annotation.

Figure 6.1 shows how interaction of feature values for `num` and `VLen` in combination with the adjective identity feature can help classify the two ANs containing adjective *large* as correct (class `1`) or incorrect (class `-1`).

In Table 6.2 we report the results for the OOC and IC annotation. The accuracy is reported with its mean ± standard deviation over the 5 data splits. We compare the DT classifier results to those obtained with the baseline system described in Chapter 4, as well as to the lower and upper bounds set as before. The results show that a classifier that uses the output of the semantic models as features outperforms the comparison-based baseline system by a large margin.

Type	Classifier accuracy	Baseline system	Lower bound	Upper bound
OOC	**0.8113** ± 0.0149	0.3810 (0.5313)	0.7889	0.8650
IC	**0.6535** ± 0.0189	0.4624 (0.4937)	0.5084	0.7467

Table 6.2: *Decision Tree* classification results on ANs

6.3.2 VO dataset

Similarly to the experiments on the AN dataset, we ran the classifier on the VO examples using the *mult* model.

We ran ablation tests and found that the best-performing feature set for the *mult* model on the OOC annotation uses a combination of four features including *verb*, `cosV`, `dens` and `OverN`, while for the IC annotation the best-performing feature set includes `VLen`, `cosV`, `RDens`, `num` and `OverV`. We note that the features that perform well with the ML classifier showed good performance in the previous experiments (see Tables 5.14 to 5.18) as well.

In Table 6.3 we report the results for the OOC and IC annotation. The accuracy is reported with its mean ± standard deviation over the 5 data splits, and the ML classifier results are compared to the results of the baseline system as well as to the lower and upper bounds.

We see that the ML classifier that uses the semantic models to derive its features outperforms the baseline system. The difference on the IC-annotated dataset, however, is only

Type	Classifier accuracy	Baseline system	Lower bound	Upper bound
OOC	**0.6577** ± 0.0166	0.4969 (0.5220)	0.5557	0.8217
IC	**0.6491** ± 0.0188	0.6048 (0.5997)	0.6086	0.8467

Table 6.3: *Decision Tree* classification results on VOs

0.0443. We see that the lower bound is itself hard to beat: for example, the lower bound for the IC annotation is slightly higher than 0.60. The classifier does not reach the upper bound, showing that there is still room for improvement.

6.4 Discussion

Our results in the previous section show that a classifier that uses the output of the semantic models as features outperforms the comparison-based baseline system and shows good accuracy. Comparison of the accuracy to the lower bound shows how well the classifier performs in identifying whether a combination is correct or incorrect. However, in §2.5 we have motivated the usefulness of reporting other evaluation measures for the ED systems. In particular, we believe that the systems should be oriented towards high precision, therefore precision of the ED algorithm should be discussed when reporting the results. Equation 2.11 in §2.5 shows that precision of less than 0.5 on the class of errors means that the system misidentifies correct use as an error more frequently than it correctly detects an error. When we discuss precision, we refer to the threshold of 0.5 as a reasonable threshold for measuring system reliability.

In this section, we analyse the classifier's performance in more detail. We report the precision and recall on the classes of correct and incorrect instances separately. Then, we also look at how the classifier performs on the different types of errors annotated in our datasets: for example, how it performs in identifying errors that are caused by semantically-related confusions or by form-related confusions.

6.4.1 AN dataset

Table 6.4 reports precision and recall for the classifier on the annotated AN dataset, as well as the F_1- and $F_{0.5}$-measures. Since we focus on precision, we mark the precision values in bold.

Our classifier achieves good precision values with respect to both OOC and IC annotation, on correct and incorrect examples. In particular, we note that P is well above 0.5 on both classes and with respect to both types of annotation. This shows that the implemented ED system is helpful in guiding a learner to text regions in need of reformulation and that it can be trusted with the instances it identifies as errors. For example, $P = 0.75$ on the instances annotated as incorrect OOC means that 3 out of 4 cases detected as errors by the system are indeed annotated as errors in our dataset.

Recall on the instances annotated as incorrect OOC is only 0.2488. This can be explained by the fact that the OOC data is highly skewed: about 79% of the ANs in this dataset

Type	Correct				Incorrect			
	P	R	F_1	$F_{0.5}$	**P**	R	F_1	$F_{0.5}$
OOC	**0.8193**	0.9762	0.8909	0.8465	**0.7500**	0.2488	0.3736	0.5346
IC	**0.6173**	0.7226	0.6658	0.6358	**0.7017**	0.5898	0.6409	0.6760

Table 6.4: Performance on correct and incorrect ANs

are correct. The classifier has many more examples of the correct instances to learn from and, as a result, has higher coverage for the class of correct instances. The classifier's performance on the IC dataset is more consistent since the dataset is more balanced.

We also investigate the performance of the classifier on the different error types. Since the classifier does not assign particular error tags to the ANs, its performance on different error types can only be measured in terms of the error rate in assigning an instance belonging to a particular error type to the appropriate class: for example, *classic dance* is annotated with `C-JF-N`, since it is correct OOC but incorrectly used IC. If this example is assigned by the classifier to the class of correct OOC and IC instances, it will boost the classifier's performance on the correct OOC instances but lower it on the incorrect IC instances. Then, if there is a sufficient number of such misclassified `C-JF-N` examples, we can say that it is problematic for the classifier to identify form-related errors.

Since the majority baseline on the ANs annotated OOC is very high (79% of ANs are correct), the classifier achieves good performance on the class of correct instances and performs worse on the class of incorrect ones. The highest error rate and the lowest accuracy within the class of correct ANs is observed on the examples annotated as `C-JF-N`: 59% of such ANs are detected as correct by the classifier, which means that the error rate on this particular group of instances is 0.41. For most other error types the error rate is 0. It is interesting to note that the highest accuracy and the lowest error rate within the class of incorrect ANs is observed on the instances annotated as `I-JF-N`: 71% of such ANs are correctly identified as errors by the classifier. This suggests that the ANs that involve form-related confusions are more readily recognised by the ED system as incorrect since it might be the case that even when they are annotated as correct OOC they still bear similarity to incorrect instances.

Among the ANs annotated as incorrect IC, the majority of ANs that are misclassified as correct by the classifier belong to the group of combinations where the error is caused by semantically related confusion (tag `S`): 70% of missed errors are in the ANs with semantically-related confusions, while only 19% of those are in the ANs with form-related confusions. The instances which are incorrect due to some confusion which cannot be explained by semantics or form relation (tag `N`) prove to be the easiest to detect – only 11% of missed errors belong to this category. However, this distribution is also subject to the original distribution of error types in our data: as Table 3.10 shows, 56.20% of all errors in ANs are due to semantically-related confusions, while 27.85% are due to form-related confusions and 15.95% are due to other reasons.

The accuracy of ED within the groups of ANs which are annotated using `S` is only 54%, while for `F` it is 69% and for `N` it is 76%. It appears that the classifier relying on semantically-motivated features misses a number of cases where the original AN and its correction are semantically similar: for example, it misses the errors in **big|great anger*, **biggest|greatest painter* and **small|short speech*. Since the ANs in these pairs are semantically similar, the features based on their semantic representations might not be discriminative enough. In contrast, the classifier is more effective in detecting errors in cases where the original AN and its correction are similar in form or not related.

Finally, Table 3.9 shows that most errors (66.50% of all errors in ANs) are due to the confusion between the correct and the chosen adjective, while 27.66% of all errors are due to the confusion between the correct and the chosen noun. At the same time, it appears that detecting an error when it is due to an incorrectly chosen adjective is easier for the classifier as it correctly identifies that there is an error 66% of the time, while it appears to be harder to identify an error if it is caused by the choice of a noun as only 47% of such cases are recognised by the classifier. The possible reason for this is that in our dataset we have a much smaller set of adjectives than nouns and the classifier learns the confusion patterns for the adjectives more reliably than for the nouns.

6.4.2 VO dataset

Table 6.5 presents the results for VO combinations. The precision values for both OOC and IC annotation and on both correct VOs and errors are higher than 0.5, with P on errors slightly higher than on the correct VOs. We conclude that the classifier can reliably detect errors in the VO datasets. Recall values for the OOC and IC annotation on the correct VOs and errors are in complementary relation to each other: high recall on the class of correct OOC instances complemented by low recall on errors shows that the classifier tends to assign more VOs to the class of correct instances and most misclassified examples are the errors missed by the classifier; at the same time, high recall on the class of incorrect IC examples with low recall on the class of correct ones means that the classifier tends to assign more VOs to the class of errors and most misclassified examples are correct VOs misclassified as errors. The majority class for the OOC annotation is represented by correct VOs (55.57% of VOs) while the majority class for the IC annotation is represented by incorrect ones (60.86% of VOs). We suppose that the changes in the recall values are due to the classifier choosing the majority class more frequently.

For the OOC annotation, the classifier performs well on the class of correct combinations, and for most error types associated with the correct OOC annotation the error rate equals 0. The highest error rate of 0.15 for the class of correct combinations is observed on the VOs annotated as `C-VN-N`. At the same time, the classifier shows poorer performance on the incorrect instances which represent the minority class for the OOC annotation: among the categories that show the worst performance are the VOs annotated as `I-VM-N` which require a change of the direct object to indirect one as in the case of **ask explanation* instead of *ask for explanation*, or **pay study* instead of *pay for study*. The error rate on this group of VOs is 0.75 which shows that most of them are misidentified as correct by the classifier. The best performance is achieved by the classifier on the VOs annotated as `I-VF-N` – the error rate on these combinations is 0.31 which is the lowest for the incorrect IC VOs. The examples of this type of VO include **rise child* instead of *raise child* and **loose contact* instead of *lose contact*.

For the IC annotation, most of the misclassified instances are the correct VOs tagged as errors by the classifier. The classifier's performance on the error type-specific subsets

Type	Correct				Incorrect			
	P	R	F_1	$F_{0.5}$	**P**	R	F_1	$F_{0.5}$
OOC	**0.6497**	0.8688	0.7434	0.6842	**0.6837**	0.3767	0.4858	0.5879
IC	**0.6027**	0.3192	0.4174	0.5118	**0.6637**	0.8630	0.7503	0.6958

Table 6.5: Performance on correct and incorrect VOs

is quite stable: among instances which are annotated with `S` tag, 83% are recognised as errors by the classifier, while for the `F`, `N` and `M` error types the recognition rate is as high as 87%, 85% and 93%, respectively. These figures show that the classifier generalises well on the incorrect instances of different subtypes, however, its performance on the correct combinations is poorer. The distribution of missed errors is as follows: 42% of those are of the type `N` (for example, **take|get hi-fi*), 32% are the errors of type `S` (**rise|increase punctuality*), 19% are of type `F` (**attain|attend course*), and the other 7% are the errors of type `M` (**learn|learn about internet*). This distribution is in accordance with the distribution of error types in the data (see Table 3.10) with the `N` type being the most frequent (37.64% of all the errors) and `M` type being the least frequent (13.90%).

Table 3.9 shows that most errors (76.28% of all errors in VOs) are due to the confusion between the correct and the chosen verb, while 19.37% are due to the confusion between the nouns. Unlike errors in the ANs discussed earlier, 71% of all the errors missed by the classifier in the VOs are errors caused by the incorrect choice of the verb, and recognition rate on such errors is 68%. At the same time, 83% of the errors caused by the incorrect choice of a noun are detected by the classifier.

6.5 Summary

We have discussed how ED in content word combinations can be cast as a binary classification task and presented an ML classifier that performs ED on the content word combinations. Our preliminary experiments showed that a *Decision Tree* classifier achieves good accuracy and precision on this task. We note that future research in this area can investigate further what type of ML classification algorithms are most appropriate for this task. The novelty of our approach is that we use the features based on measures of semantic anomaly described in Chapter 5 and the total number of features is 14.

Our experiments have confirmed that features derived from the semantic measures perform well on this task. In particular, measures based on the neighbourhood of the model-generated vectors result in features that show the highest performance. The particular ranking of the features and the order of their application to the classification task are defined by the algorithm and the interaction of features. We have applied feature binning to convert the real-valued features into binary features to reduce the sparsity of the feature space and boost informativeness of individual features.

We have obtained good accuracy on all annotated datasets for both ANs and VOs. The results presented in §6.3 show that the classifier based on semantic features outperforms the algorithms that have previously been applied to this task (see Chapter 4) by a large margin. We note that the DT classifier using semantic features beats the baseline which for this task is quite high. However, we also note that the classifier's performance does not reach the estimated upper bound. In particular, the upper bound on the OOC and IC annotation of the VO examples is rather high (0.8217 and 0.8467 respectively) and the classifier's accuracy is about 0.2 below it. This shows that there is room for improvement on the results presented in this work.

In §6.4, we analysed the performance of the classifier on the ANs and VOs in more detail. We discussed why it is important to focus on precision of the ED algorithms and reported precision, recall and F-measures on the classes of correct and incorrect instances separately. We have also discussed the cases that are most difficult for the classifier to detect: for example, a significant number of missed errors in the AN dataset belong to the `S` error type where the chosen incorrect word is semantically related to the correct one. We conclude that a classifier that uses semantic information to derive features might perform worse on the cases where the confusion is caused by semantic similarity.

Chapter 7

Conclusions

7.1 Contributions

This thesis has addressed the task of error detection in content word combinations, which is to date an under-explored area in learner language research. The previous research in error detection and correction has mainly focused on other types of errors, or performed error correction rather than detection. We have focused on adjective–noun and verb–object combinations. Our analysis of learner data has shown that these types of combinations cover a substantial portion of learner errors in the use of content words, and we hope that the results presented in this work will motivate the use of similar approaches to address error detection in other types of content word combinations.

In Chapter 2, we reviewed the field of error detection and correction in learner data. The tasks related to learner language have attracted much attention in recent years, but the research has primarily focused on detection and correction of grammatical errors and errors in function words. We show that errors in content words should be addressed with different methods and that they are challenging for existing algorithms. We also show that state-of-the-art approaches to error detection in content word combinations have a number of limitations, and the current work aims to address this gap in the field.

Content word combinations allow higher variability and, in general, there are no clear-cut rules of English that dictate the correct use of content words. In this work, we mainly focus on error detection since we believe that learners benefit most from it and should be notified of the incorrectly chosen content words in their writing. We argue that error correction can be performed in an interactive way when the learner can be presented with possible alternatives to incorrect word combinations and allowed to make the final choice of correction.

The relative lack of attention to errors in content word combinations has resulted in an absence of thorough analysis of the typical errors committed by learners, as well as datasets exemplifying such errors that can be used by the NLP community. In this work, we have collected and presented a dataset of errors in AN and VO combinations that are not attested in a native corpus of English. These combinations, on the one hand, illustrate typical confusion patterns, and, on the other hand, are challenging for existing error detection algorithms which are based on the idea that “correctness” in content word combinations can be equated with higher fluency. We have devised an annotation scheme for the datasets which allows us to annotate the combinations with respect to their correctness and the possible source of error. The datasets of 798 AN combinations and 800 VO combinations are publicly available at `http://www.ilexir.com/` and `http://www.cl.cam.ac.uk/~ek358/data/` together with the annotation scheme and annotation guidelines. We believe that the annotation scheme that we devised for these datasets can easily be applied to other types of content word combinations, or extended if necessary. Chapter 3 discusses collection and annotation of the data.

In Chapter 4 we described a simple algorithm for detecting errors in content word combinations based on the idea that more fluent combinations should be chosen over less fluent ones since they are more "correct". This approach has previously been applied to error detection in content words, and it is mainly aimed at writing improvement rather than error detection *per se*. We showed the limitations of this approach and, in particular, discussed why it would not help detecting errors in learner writing which contains a substantial number of rare or corpus-unattested combinations. We maintain that approaches to error detection which are based on modelling the meaning of the combinations using compositional distributional semantics are more promising than approaches based on comparison of corpus occurrence counts.

Chapter 5 discusses implementation of models of compositional distributional semantics and their application to error detection in content words. To the best of our knowledge, this is the first work which treats the task of error detection with models based on semantics. The novelty of this work is that we not only show that semantic models can be applied to this task, but we also explore the properties of the model-generated representations that highlight the differences between correct and incorrect combinations.

Finally, we have also shown how to cast the task of error detection in content words as a binary classification problem. In Chapter 6 we presented and discussed the implementation of a machine learning classifier which uses features derived from semantic representations. We showed that this classifier achieves high accuracy and precision on the datasets of AN and VO combinations and outperforms the previous approaches to this task.

7.2 Directions for future research

In §2.3.2 we presented a three-step algorithm that describes error detection and correction in content words. We also discussed that previous approaches mainly focused on steps two (search for alternatives) and three (error correction) skipping step one (error detection), or merged the steps, making error detection dependent on the set of alternatives and, thus, on correction. However, we believe that error detection is an important step that should be performed independently of error correction and refer to Leacock et al. (2009) who showed that a mere identification of the error location is often enough for the learners to rewrite the text and correct the errors themselves. In this work, we have focused on step one of the EDC algorithm, and our future research will address the subsequent steps related to error correction.

Step two of the EDC algorithm is concerned with the search for possible alternatives. The alternatives can be found among the set of related words, such as synonyms and hypo-

/hypernyms (see Chapter 4), or nearest neighbours in the semantic space (see Chapter 5). The majority of the errors in our annotated datasets are caused by confusion between semantically related words or words spelled or pronounced similarly, so that the confusion set consisting of such related words would cover a substantial amount of error cases (up to 85% of the errors in ANs and up to half of the errors in VOs in our datasets). Manually created resources such as WordNet can provide us with high-quality data, but they might not cover all possible related alternatives, and distributional semantics can help overcome the bottleneck. The recent studies of Cahill et al. (2013) and Madnani and Cahill (2014) also show that one can use high-quality manual resources like Wikipedia with its submitted revisions to collect possible corrections on a large scale.

In step three, the collected alternatives should be assessed and the most appropriate one should be suggested to the learner as a correction. In §2.3.2 as well as in §4 we explained that previous approaches used metrics based on frequency of occurrence to choose the correction, but we maintain that (i) "correctness" should not be equated with fluency, and (ii) unlike error detection which should be performed on learner text in an unambiguous manner, error correction should be treated as an open-ended task. Previous research (Chodorow et al., 2010; Andersen et al., 2013) has shown that learners are able to make informed decisions about the appropriate corrections when presented with possible alternatives. Since content words express meaning, the appropriate correction depends on the communicative intent of the learner. For example, an ED algorithm may identify that **big conversation* is incorrect, but a correction algorithm has no means of deciding whether the learner meant *important conversation*, *great conversation* or *long conversation*. We believe that error correction should be implemented as an interactive process where the learner is presented with a number of possible alternatives, with some supporting examples and explanations, from which they can choose the one that fulfils their communicative goal.

We believe that an error correction algorithm should not focus on finding a single best correction but rather provide the learner with a set of alternatives. It has been shown before that many contexts license several alternative corrections even for function words and the performance of the systems that are assessed on the basis of a single correction is underestimated. For example, Tetreault and Chodorow (2008a) focused on the use of prepositions and showed that when only a single correction was allowed, the inter-annotator agreement was only about 76%. Lee et al. (2009) found that annotators often identified more than one possible correct construction in an experiment on the use of articles and noun number, and according to their experiment, an EDC system's performance may be underestimated by 18% or more if multiple possible corrections are not taken into account. Our example with **big conversation* illustrates the need for multiple corrections for content words, and an advantage of the interactive error correction algorithm that we propose here is that it can provide the learner with a *set* of corrections. We note, however, that the ability to make informed choices may depend on learners' language proficiency, and the extent to which students at the lower levels of proficiency are able to choose the appropriate corrections should be taken into account.

We have shown that a machine learning classifier using features based on compositional distributional representations achieves good results. However, since the results do not reach the upper bound, future research should investigate ways to improve and extend the implementation of the algorithm at different stages, such as:

- *Semantic space construction*: The way the semantic space is set has a direct effect on the output of the semantic models. For example, we use *LMI* as the weighting scheme and *SVD* for space reduction, which can both produce negative values. Since the semantic space has a direct geometric interpretation, negative values can be interpreted as coordinates in the geometric space defining the direction of the distributional vector. At the same time, positive values in the dimensions of the distributional vectors are easier to interpret. Recent work (Lazaridou et al., 2013; Vecchi, 2013) has used *Positive Pointwise Mutual Information* for weighting and *Non-negative Matrix Factorisation* for dimensionality reduction. Vecchi (2013) reported that these methods produce a semantic space of better quality, which has been confirmed by the results in the semantic similarity task. We conclude that further experiments are needed to verify whether a different setting of the semantic space has an effect on the results in our task.

- *Different models of compositional semantics*: The best results in our experiments were obtained with the simpler models of semantic composition. Future research can investigate the application of other models, such as *w.add* (weighted additive) and *dl* (dilation), which have been reported to outperform simpler models in other tasks (Lazaridou et al., 2013; Vecchi, 2013).

- *Additional measures for detecting semantic mismatch*: The ED algorithm in our experiments relies on the set of 14 features based on semantic representations. Future research should investigate ways to extend the list of measures that can capture the difference between semantic representations of correct and incorrect combinations. Such measures can be informed by research on lexical and compound processing in cognitive science and psycholinguistics: for example, it has been shown that the *frequency* of occurrence of the constituents within compounds has an effect on lexical processing (Andrews et al., 2009; Juhasz et al., 2003; Pollatsek et al., 2000), and it can be assumed that more frequent constituents generally produce more acceptable combinations than rarer constituents. *Family size* measured as the number of distinct words (for example, nouns) a given modifier (adjective) can be seen to modify in a corpus also has an effect on word combination processing times, and it can be assumed that highly productive words correspond to more flexible semantics and should be found more often in acceptable ANs (Vecchi, 2013).

In the current implementation, we have applied a rather generic ED algorithm that does not distinguish between different types of erroneous words. At the same time, we have noted that errors in content words are more diverse and less systematic than those in function words, and distinguishing between different types of adjectives or verbs might prove to be helpful for ED. For example, Rozovskaya et al. (2014) have applied a linguistically-motivated approach to grammatical verb error correction that makes use of the notion of verb finiteness to identify triggers and types of mistakes. Boleda et al. (2012) and Vecchi (2013) have looked into different types of adjectival modification, distinguishing between *intersective* use of colour terms such as *white dress*, *subsective* use of colour terms such as *white wine*, and *intensional* use of adjectives such as *former wife*. Vecchi (2013) shows that semantic treatment for these different types of adjectival modification needs to be differentiated. On the one hand, the different types of adjectives and verbs can trigger different types of errors, and on the other, they can contribute differently to the semantic representations of the ANs and VOs. Future research can further investigate ways to use information about different types of adjectives and verbs in ED.

Another factor that is not directly modelled by the semantic representations used in this work is the effect of the particular context on the ANs and VOs. We have noted that the semantic models applied in this work can model the general representation for the word combinations, and we evaluate the ED approach on the *type-based* annotation (see the discussion in §3.1). Future work can investigate ways to introduce context-sensitive information into semantic models. One promising approach is to use topic coherence which shows the semantic relatedness of the items in a given set of words (Steyvers and Griffiths, 2007; Newman et al., 2010), and therefore, can be assumed to drop in a context where an incorrectly chosen word is used. The introduction of context-sensitive information in semantic models has also been considered by Erk and Padó (2010), Reisinger and Mooney (2010), and Thater et al. (2011).

In recent years, context-predicting models, commonly referred to as embeddings or neural language models, have gained much popularity (Bengio et al., 2003; Collobert and Weston, 2008; Turian et al., 2010; Collobert et al., 2011; Huang et al., 2012; Mikolov et al., 2013). Within this type of model, the weights in a word vector are assigned so as to maximise the probability of the contexts in which the word is observed in the corpus rather than set, relying on various criteria, on the word vectors constructed using the contexts (Baroni et al., 2014b). Baroni et al. (2014b) have performed a thorough comparison of the performance of distributional semantic models built in the traditional way and those based on context prediction and shown that the latter outperform the former in a number of tasks including semantic relatedness, synonym detection and analogy. It is interesting to note that the only task in which the models set in the traditional way (Herdağdelen and Baroni, 2009; Baroni and Lenci, 2010) outperformed the models based on word embeddings is selectional preference detection, which is close to the task we address in this work. However, given the success of the models based on word embeddings on other tasks, we conclude that future work should investigate the implementation of such models and perform a comparison to the system used in this work.

Finally, we plan to investigate the application of the approach presented in this thesis to ED in other types of content word combinations.

Bibliography

Ø. Andersen, J. Nioche, T. Briscoe, and J. Carroll. The BNC parsed with RASP4UIMA. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pages 865–869, 2008.

Ø. Andersen, H. Yannakoudakis, F. Barker, and T. Parish. Developing and testing a self-assessment and tutoring system. In *Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2013)*, pages 32–41, 2013.

L. Andersson and P. Trudgill. *Bad Language.* Oxford: Basil Blackwell, 1990.

A.-M. Andreasson. Norm as a pedagogical paradigm. *World Englishes*, 13(3):395–409, 1994.

M. Andrews, G. Vigliocco, and D. Vinson. Integrating experiential and distributional data to learn semantic representations. *Psychological Review*, 116:463–498, 2009.

G. Aston. *Corpora in language pedagogy: Matching theory and practice. In G. Cook & B. Seidlhofer (eds.), Principle and Practice in Applied Linguistics: Studies in Honour of H.G. Widdowson*, pages 257–270. Oxford: Oxford University Press, 1995.

Y. Attali. Exploring the feedback and revision features of the Criterion service. In *Proceedings of the Annual Meeting of the National Council on Measurement in Education (NCME)*, pages 1–22, 2004.

M. Baroni and A. Lenci. Distributional Memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, 2010.

M. Baroni and R. Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 1183–1193, 2010.

M. Baroni, R. Bernardi, and R. Zamparelli. Frege in space: A program for compositional distributional semantics. *Linguistic Issues in Language Technologies*, 9(6):5–110, 2014a.

M. Baroni, G. Dinu, and G. Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 238–247, 2014b.

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

M. Benson. Collocations and general-purpose dictionaries. *International Journal of Lexicography*, 3(1):23–25, 1990.

S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit.* Sebastopol, CA: O'Reilly Media, 2009.

J. Bitchener. Evidence in support of written corrective feedback. *Journal of Second Language Writing*, 17(2):102–118, 2005.

J. Bitchener, S. Young, and D. Cameron. The effect of different types of corrective feedback on ESL student writing. *Journal of Second Language Writing*, 14(3):191–205, 2008.

W. Blacoe and M. Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, pages 546–556, 2012.

G. Boleda, E. Vecchi, M. Cornudella, and L. McNally. First-order vs. higher-order modification in distributional semantics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, pages 1223–1233, 2012.

E. Briscoe, J. Carroll, and R. Watson. The Second Release of the RASP System. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006) Interactive Presentation Sessions*, pages 59–68, 2006.

C. Brockett, W. B. Dolan, and M. Gamon. Correcting ESL Errors Using Phrasal SMT Techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 249–256, 2006.

J. Brooke and G. Hirst. Native language detection with 'cheap' learner corpora. In *Proceedings of the 2011 Conference on Learner Corpus Research (LCR 2011)*, pages 37–47, 2011.

J. Brooke and G. Hirst. Measuring Interlanguage: Native Language Identification with L1-influence Metrics. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 779–784, 2012.

J. A. Bullinaria and J. P. Levy. Extracting Semantic Representations from Word Co-occurrence Statistics: A computational study. *Behavior Research Methods*, 39:510–526, 2007.

J. A. Bullinaria and J. P. Levy. Extracting Semantic Representations from Word Co-occurrence Statistics: Stop-lists, Stemming and SVD. *Behavior Research Methods*, 44: 890–907, 2012.

L. Burnard. The British National Corpus, version 3 (BNC XML Edition). Distributed by Oxford University Computing Services on behalf of the BNC Consortium. `http://www.natcorp.ox.ac.uk/`, 2007.

A. Cahill, N. Madnani, J. Tetreault, and D. Napolitano. Robust Systems for Preposition Error Correction Using Wikipedia Revisions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, pages 507–517, 2013.

Y. C. Chang, J. S. Chang, H. J. Chen, and H. C. Liou. An automatic collocation writing assistant for Taiwanese EFL learners: A case of corpus-based NLP technology. *Computer Assisted Language Learning*, 21(3):283–299, 2008.

M. Chodorow, M. Gamon, and J. Tetreault. The utility of grammatical error detection systems for English language learners: Feedback and Assessment. *Language Testing*, 27(3):335–353, 2010.

M. Chodorow, M. Dickinson, R. Israel, and J. Tetreault. Problems in Evaluating Grammatical Error Detection Systems. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 611–628, 2012.

N. Chomsky. *Syntactic Structures.* The Hague: Mouton, 1957.

N. Chomsky. *Knowledge of Language: Its Nature, Origin, and Use.* New York: Praeger, 1986.

S. Clark. *Vector space models of lexical meaning. In S. Lappin and C. Fox (eds.), Handbook of Contemporary Semantics, 2nd ed.*, chapter 16. Malden, MA: Wiley-Blackwell. In press. `http://www.cl.cam.ac.uk/~sc609/pubs/sem_handbook.pdf`, 2015.

J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, pages 160–167, 2008.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12: 2493–2537, 2011.

V. J. Cook. *Second Language Learning and Language Teaching.* London: Edward Arnold, 1991.

S. P. Corder. Idiosyncratic dialects and error analysis. *IRAL: International Review of Applied Linguistics in Language Teaching*, 9(2):147–160, 1971.

S. P. Corder. *Error analysis. In J. P. Allen and S. P. Corder (eds.), Techniques in Applied Linguistics (The Edinburgh Course in Applied Linguistics)*, volume 3. London: Oxford University Press, 1974.

A. P. Cowie and P. Howarth. Phraseological Competence and Written Proficiency. Paper Presented at the British Association of Applied Linguistics Conference (BAAL), 1995.

D. Crystal. *English as a global language.* Cambridge: Cambridge University Press, 2nd edition, 2003.

E. Dagneaux, S. Denness, and S. Granger. Computer-aided error analysis. *System*, 26: 163–174, 1998.

D. Dahlmeier and H. T. Ng. Correcting Semantic Collocation Errors with L1-induced Paraphrases. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 107–117, 2011a.

D. Dahlmeier and H. T. Ng. Grammatical error correction with alternating structure optimization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, volume 1, pages 915–923, 2011b.

D. Dahlmeier, H. T. Ng, and S. M. Wu. Building a large annotated corpus of learner English: The NUS corpus of learner English Authors. In *Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2013)*, pages 22–31, 2013.

R. Dale and A. Kilgarriff. Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG 2011)*, volume Helping Our Own: The HOO 2011 Pilot Shared Task, pages 242–249, 2011.

R. Dale, I. Anisimoff, and G. Narroway. HOO 2012: A Report on the Preposition and Determiner Error Correction Shared Task. In *Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2012)*, pages 54–62, 2012.

G. M. Dalgish. Computer-assisted ESL research. *CALICO Journal*, 2(2):32–37, 1985.

H. C. Dulay, M. K. Burt, and S. D. Krashen. *Language two.* Oxford: Oxford University Press, 1982.

R. Ellis. *The Study of Second Language Acquisition.* Oxford: Oxford University Press, 1994.

R. Ellis, Y. Sheen, M. Murakami, and H. Takashima. The effects of focused and unfocused written corrective feedback in an English as a foreign language context. *System*, 36(3): 353–371, 2008.

K. Erk and S. Padó. A Structured Vector Space Model for Word Meaning in Context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 897–906, 2008.

K. Erk and S. Padó. Exemplar-based models for word meaning in context. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010): Short Papers*, pages 92–97, 2010.

S. Evert. *The Statistics of Word Cooccurrences.* PhD thesis, Stuttgart University, 2005.

M. Felice and Z. Yuan. Generating artificial errors for grammatical error correction. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pages 116–126, 2014.

M. Felice, Z. Yuan, Ø. Andersen, H. Yannakoudakis, and E. Kochmar. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL 2014): Shared Task*, pages 15–24, 2014.

R. De Felice and S. G. Pulman. A classifier-based approach to preposition and determiner error correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, volume 1, pages 169–176, 2008.

J. R. Firth. *A synopsis of linguistic theory 1930-1955. In Studies in Linguistic Analysis*, pages 1–32. Oxford: Philological Society, 1957.

P. Flach. *Machine Learning: The Art and Science of Algorithms that Make Sense of Data.* Cambridge: Cambridge University Press, 2012.

J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382, 1971.

P. W. Foltz, W. Kintsch, and T. K. Landauer. The Measurement of Textual Coherence with Latent Semantic Analysis. *Discourse Processes*, 25:285–307, 1998.

G. Frege. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50, 1892.

H. Frei. *La grammaire des fautes.* Paris: Geuthner, 1929.

Y. Futagi, P. Deane, M. Chodorow, and J. Tetreault. A computational approach to detecting collocation errors in the writing of non-native speakers of English. *Computer Assisted Language Learning*, 21(4):353–367, 2008.

M. Gamon. Using Mostly Native Data to Correct Errors in Learners' Writing: A Meta-Classifier Approach. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010)*, pages 163–171, 2010.

M. Gamon, J. Gao, C. Brockett, A. Klementiev, W. Dolan, D. Belenko, and L. Vanderwende. Using Contextual Speller Techniques and Language Modeling for ESL Error Correction. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pages 449–456, 2008.

S. Granger. *A Bird's-eye View of Computer Learner Corpus Research. In Teubert W. and Krishnamurthy R. (eds.), Corpus Linguistics: Critical Concepts in Linguistics*, volume 2, pages 44–72. London: Routledge, 2007.

S. Granger and G. Leech. *Learner English on Computer (Studies in Language and Linguistics).* London: Routledge, 1998.

E. Grefenstette. Towards a Formal Distributional Semantics: Simulating Logical Calculi with Tensors. In *Proceedings of *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*, pages 1–10, 2013.

E. Grefenstette and M. Sadrzadeh. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 1394–1404, 2011.

E. Grefenstette, G. Dinu, Y.-Z. Zhang, M. Sadrzadeh, and M. Baroni. Multi-step regression learning for compositional distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, pages 131–142, 2013.

E. Guevara. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics (GEMS 2010)*, pages 33–37, 2010.

S. Gui and H. Yang. Computer Analysis of Chinese Learner English. `http://lc.ust.hk/~center/conf2001/keynote/subsect4/yang.pdf`, 2001.

Y. Guo and G. H. Beckett. The Hegemony of English as a Global Language: Reclaiming Local Knowledge and Culture in China. *Convergence*, 40(1/2):117, 2007.

B. Hammarberg. The Insufficiency of Error Analysis. *IRAL: International Review of Applied Linguistics in Language Teaching*, 12(1–4):185–192, 1974.

N.-R. Han, J. R. Tetreault, S.-H. Lee, and J.-Y. Ha. Using an Error-Annotated Learner Corpus to Develop an ESL/EFL Error Correction System. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 763–770, 2010.

Z. Harris. Distributional structure. *Word*, 10(2/3):146–162, 1954.

G. E. Heidorn, K. Jensen, L. A. Miller, R. J. Byrd, and M. S. Chodorow. The EPISTLE text-critiquing system. *IBM Systems Journal*, 21(3):305–326, 1982.

A. Helfrich and B. Music. Design and Evaluation of Grammar Checkers in Multiple Languages. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, pages 1036–1040, 2000.

A. Herdağdelen and M. Baroni. BagPack: A general framework to represent semantic relations. In *Proceedings of the 2009 Workshop on GEometrical Models of Natural Language Semantics (GEMS 2009)*, pages 33–40, 2009.

M. Hermet, A. Désilets, and S. Szpakowicz. Using the Web as a Linguistic Resource to Automatically Correct Lexico-Syntactic Errors. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pages 390–396, 2008.

E. Huang, R. Socher, C. Manning, and A. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 873–882, 2012.

C. James. *Errors in Language Learning and Use: Exploring Error Analysis.* London: Longman, 1998.

B. Juhasz, M. Starr, A. Inhoff, and L. Placke. The effects of morphology on the processing of compound words: Evidence from naming, lexical decisions and eye fixations. *British Journal of Psychology*, 94:223–244, 2003.

D. Kiela and S. Clark. A Systematic Study of Semantic Vector Space Model Parameters. In *Proceedings of EACL 2014 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC 2014)*, pages 21–30, 2014.

A. Kilgarriff. Googleology is Bad Science. *Computational Linguistics*, 33(1):147–151, 2007.

W. Kintsch. Predication. *Cognitive Science*, 25:173–202, 2001.

G. Kjellmer. *A mint of phrases. In K. Aijmer & B. Altenberg (eds.), English Corpus Linguistics: Studies in Honour of Jan Svartvik*, pages 111–127. Harlow, Essex: Longman, 1991.

E. Kochmar. Identification of a Writer's Native Language by Error Analysis. Master's thesis, University of Cambridge, 2011.

E. Kochmar and T. Briscoe. Capturing Anomalies in the Choice of Content Words in Compositional Distributional Semantic Space. In *Proceedings of the 9th Recent Advances in Natural Language Processing Conference (RANLP 2013)*, pages 365–372, 2013.

E. Kochmar and T. Briscoe. Detecting Learner Errors in the Choice of Content Words Using Compositional Distributional Semantics. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014): Technical Papers*, pages 1740–1751, 2014.

E. Kochmar, Ø. Andersen, and T. Briscoe. HOO 2012 Error Recognition and Correction Shared Task: Cambridge University Submission Report. In *Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2012)*, pages 242–250, 2012.

R. Lahav. The combinatorial-connectionist debate and the pragmatics of adjectives. *Pragmatics and Cognition*, 1:71–88, 1993.

T. Landauer and S. Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

J. R. Landis and G. G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174, 1977.

G. Lapesa and S. Evert. Evaluating neighbor rank and distance measures as predictors of semantic priming. In *Proceedings of the 4th Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2013)*, pages 66–74, 2013.

A. Lazaridou, E. Vecchi, and M. Baroni. Fish transporters and miracle homes: How compositional distributional semantics can help NP parsing. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1908–1913, 2013.

C. Leacock and M. Chodorow. *Automated Grammatical Error Detection. In M. D. Shermis and J. C. Burstein (eds.), Automated Essay Scoring: A Cross-Disciplinary Perspective*, pages 195–207. Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

C. Leacock, M. Gamon, and C. Brockett. User Input and Interactions on Microsoft Research ESL Assistant. In *Proceedings of the 4th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2009)*, pages 73–81, 2009.

C. Leacock, M. Chodorow, and J. Tetreault. *Automated Grammatical Error Detection for Language Learners.* Morgan & Claypool Publishers, 2010.

C. Leacock, M. Chodorow, M. Gamon, and J. Tetreault. *Automated Grammatical Error Detection for Language Learners.* Morgan & Claypool Publishers, second edition, 2014.

J. Lee, J. Tetreault, and M. Chodorow. Human Evaluation of Article and Noun Number Usage: Influences of Context and Construction Variability. In *Proceedings of the 3rd Linguistic Annotation Workshop (LAW 2009)*, pages 60–63, 2009.

A. Lenci. Distributional approaches in linguistic and cognitive research. *Italian Journal of Linguistics*, 20(1):1–31, 2008.

A. A. Lipnevich and J. K. Smith. Response to Assessment Feedback: The Effects of Grades, Praise, and Source of Information. Technical report, Educational Testing Service, 2008.

A. L.-E. Liu. A corpus-based lexical semantic investigation of verb-noun miscollocations in Taiwan learners English. Master's thesis, Tamkang University, Taipei, 2002.

A. L.-E. Liu, D. Wible, and N.-L. Tsao. Automated suggestions for miscollocations. In *Proceedings of the 4th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2009)*, pages 47–50, 2009.

A. Lüdelig, M. Walter, E. Kroymann, and P. Adolphs. Multi-level error annotation in learner corpora. In *Proceedings of the Corpus Linguistics 2005 Conference*, 2005.

J. Lyons. *Introduction to Theoretical Linguistics.* Cambridge: Cambridge University Press, 1968.

N. Macdonald, L. Frase, P. Gingrich, and S. Keenan. The Writer's Workbench: Computer Aids for Text Analysis. *IEEE Transactions on Communications*, 30(1):105–110, 1982.

N. Madnani and A. Cahill. An Explicit Feedback System for Preposition Errors based on Wikipedia Revisions. In *Proceedings of the 9th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2014)*, pages 79–88, 2014.

N. Madnani, J. R. Tetreault, M. Chodorow, and A. Rozovskaya. They Can Help: Using Crowdsourcing to Improve the Evaluation of Grammatical Error Detection Systems. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pages 508–513, 2011.

S. McDonald and M. Ramscar. Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society (CogSci 2001)*, pages 611–616, 2001.

I. A. Mel'čuk. *Collocations and Lexical Functions. In A.P. Cowie (ed.), Phraseology. Theory, Analysis, and Applications*, pages 23–53. Oxford: Clarendon Press, 1998.

B. Mevik and R. Wehrens. The pls package: Principal component and partial least squares regression in R. *Journal of Statistical Software*, 18(2):1–24, 2007.

J. Mey. *Right or wrong, my native speaker. In F. Coulmas (ed.), A festschrift for native speaker*, pages 69–84. The Hague: Mouton, 1981.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. `http://arxiv.org/abs/1301.3781/`, 2013.

J. Mitchell and M. Lapata. Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-HLT 2008)*, pages 236–244, 2008.

J. Mitchell and M. Lapata. Language Models Based on Semantic Composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 430–439, 2009.

J. Mitchell and M. Lapata. Composition in distributional models of semantics. *In Cognitive Science*, pages 1388–1429, 2010.

S. Mohammad and G. Hirst. Distributional Measures as Proxies for Semantic Distance: A Survey. `http://www.umiacs.umd.edu/~saif/WebPages/publications.html`, 2007.

R. Montague. Universal grammar. *Theoria*, 36:373–398, 1970.

R. Montague. *The proper treatment of quantification in ordinary English. In K. J. J. Hintikka, J. M. E. Moravcsik, and P. Suppes (eds.), Approaches to Natural Language (Synthese Library 49)*, pages 221–242. Dordrecht: Reidel, 1973.

B. Murphy, P. Talukdar, and T. Mitchell. Selecting corpus-semantic models for neurolinguistic decoding. In *Proceedings of *SEM 2013: The First Joint Conference on Lexical and Computational Semantics*, volume 1, pages 114–123, 2012.

R. Nagata and K. Nakatani. Evaluating performance of grammatical error detection to maximize learning effect. In *Proceedings of the 23th International Conference on Computational Linguistics (COLING 2010)*, pages 894–900, 2010.

I. S. P. Nation. *Learning vocabulary in another language.* Cambridge: Cambridge University Press, 2001.

J. R. Nattinger and J. D. DeCarrico. *Lexical phrase and language teaching.* Oxford: Oxford University Press, 2001.

N. Nesselhauf. The use of collocations by advanced learners of English and some implications for teaching. *Applied Linguistics*, 24:223–242, 2003.

N. Nesselhauf. *Learner corpora and their potential for language teaching. In J. Sinclair (ed.), How to Use Corpora in Language Teaching*, pages 125–152. Amsterdam: John Benjamins, 2004.

D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010)*, pages 100–108, 2010.

H. T. Ng, S. M. Wu, Y. Wu, C. Hadiwinoto, and J. Tetreault. The CoNLL-2013 Shared Task on Grammatical Error Correction. In *Proceedings of the 17th Conference on Computational Natural Language Learning: Shared Task (CoNLL-2013 Shared Task)*, pages 1–12, 2013.

H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant. The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of the 18th Conference on Computational Natural Language Learning: Shared Task (CoNLL-2014 Shared Task)*, pages 1–14, 2014.

D. Nicholls. The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003 Conference*, pages 572–581, 2003.

G. Nunberg, T. Wasow, and I. A. Sag. Idioms. *Language*, 70(3):491–538, 1994.

R. Östling and O. Knutsson. A corpus-based tool for helping writers with Swedish collocations. In *Proceedings of the Workshop on Extracting and Using Constructions in NLP*, pages 28–33, 2009.

S. Padó and M. Lapata. Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199, 2007.

P. Pantel. Inducing ontological co-occurrence vectors. In *Proceedings of the 43rd Conference of the Association for Computational Linguistics (ACL 2005)*, pages 125–132, 2005.

T. Park, E. Lank, P. Poupart, and M. Terry. Is the sky pure today? AwkChecker: an assistive tool for detecting and correcting collocation errors. In *Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST 2008)*, pages 121–130, 2008.

A. Pollatsek, J. Hyönä, and R. Bertram. The role of morphological constituents in reading Finnish compound words. *Journal of Experimental Psychology: Human Perception and Performance*, 26(2):820–833, 2000.

N. A. Pravec. Survey of learner corpora. *ICAME Jornal*, 26:81–114, 2002.

S. Pulman. *Distributional Semantic Models. In In C. Heunen M. Sadrzadeh and E. Grefenstette (eds.), Quantum Physics and Linguistics: A Compositional Diagrammatic Discourse*, pages 333–358. Oxford: Oxford University Press, 2013.

J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 1986.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2014.

M. A. Ramos, L. Wanner, O. Vincze, G. Casamayor del Bosque, N. Vázquez Veiga, E. Mosqueira Suárez, and S. Prieto González. Towards a Motivated Annotation Schema of Collocation Errors in Learner Corpora. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 3209–3214, 2010.

R. Rapp. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the 9th Machine Translation Summit (MT Summit)*, pages 315–322, 2003.

J. Reisinger and R. J. Mooney. Multi-prototype vector-space models of word meaning. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010)*, pages 109–117, 2010.

S. D. Richardson and L. C. Braden-Harder. The experience of developing a large-scale natural language text processing system: CRITIQUE. In *Proceedings of the 2nd Conference on Applied Natural Language Processing (ANLC 1988)*, pages 195–202, 1988.

S. Z. Riehemann. *A Constructional Approach to Idioms and Word Formation.* PhD thesis, Stanford University, 2001.

A. Rozovskaya and D. Roth. Annotating ESL Errors: Challenges and Rewards. In *Proceedings of the 5th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2010)*, pages 28–36, 2010a.

A. Rozovskaya and D. Roth. Generating Confusion Sets for Context-Sensitive Error Correction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 961–970, 2010b.

A. Rozovskaya and D. Roth. Algorithm Selection and Model Adaptation for ESL Correction Tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pages 924–933, 2011.

A. Rozovskaya, D. Roth, and V. Srikumar. Correcting Grammatical Verb Errors. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pages 358–367, 2014.

H. Rubenstein and J. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

G. Ruge. Experiments on linguistically-based term associations. *Information Processing and Management*, 28(3):317–332, 1992.

M. Sahlgren. The Distributional Hypothesis. *Rivista di Linguistica*, 20(1):33–53, 2008.

H. Schütze. *Ambiguity Resolution in Natural Language Learning.* Stanford, CA: CSLI Publications, 1997.

H. Schütze and J. Pedersen. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR 1995)*, pages 161–175, 1995.

L. Selinker. Interlanguage. *International Review of Applied Linguistics*, 10:209–241, 1972.

L. Selinker. *Rediscovering Interlanguage.* London: Longman, 1992.

Y. Sheen. The Effect of Focused Written Corrective Feedback and Language Aptitude on ESL Learners' Acquisition of Articles. *TESOL Quarterly*, 41(2):255–283, 2007.

C. C. Shei and H. Pain. An ESL Writer's Collocation Aid. *Computer Assisted Language Learning*, 13(2):167–182, 2000.

J. Sinclair. *Corpus concordance collocation.* Oxford: Oxford University Press, 1991.

M. Steyvers and T. Griffiths. *Probabilistic topic models. In T. Landauer, D. McNamara, S. Dennis, and W. Kintsch (eds.), Latent Semantic Analysis: A Road to Meaning.* Mahwah, NJ: Lawrence Erlbaum Associates, 2007.

J. Tetreault and M. Chodorow. Native judgments of non-native usage: experiments in preposition error detection. In *Proceedings of the COLING 2008 Workshop on Human Judgements in Computational Linguistics*, pages 24–32, 2008a.

J. Tetreault and M. Chodorow. The ups and downs of preposition error detection in ESL writing. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, volume 1, pages 865–872, 2008b.

J. Tetreault and M. Chodorow. Examining the Use of Region Web Counts for ESL Error Detection. In *Proceedings of the 5th Web as Corpus Workshop (WAC5)*, pages 71–78, 2009.

J. Tetreault, M. Chodorow, and N. Madnani. Bucking the trend: improved evaluation and annotation practices for ESL error detection systems. *Language Resources and Evaluation*, 48(1):5–31, 2013.

S. Thater, H. Fürstenau, and M. Pinkal. Word Meaning in Context: A Simple and Effective Vector Model. In *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pages 1134–1143, 2011.

Y. Tono. Learner corpora: design, development and applications. In *Proceedings of the Corpus Linguistics 2003 Conference. University Centre for Computer Corpus Research on Language (UCREL) technical paper number 16*, pages 800–809, 2003.

J. Truscott. The Case Against Grammar Correction in L2 Writing Classes. *Language Learning*, 46(2):327–369, 1996.

J. Turian, L.-A. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 384–394, 2010.

P. Turney. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research (JAIR)*, 44:533–585, 2012.

E. Vecchi. *Distributional semantic phrases vs. semantic distributional nonsense: Adjective Modification in Compositional Distributional Semantics.* PhD thesis, Center for Brain and Mind Sciences (CIMeC), Università di Trento, Rovereto (TN) Italy, 2013.

E. Vecchi, M. Baroni, and R. Zamparelli. (Linear) maps of the impossible: Capturing semantic anomalies in distributional space. In *Proceedings of the Distributional Semantics and Compositionality (DISCO) Workshop at the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, pages 1–9, 2011.

H. Wible, C.-H. Kwo, N.-L. Tsao, A. Liu, and H.-L. Lin. Bootstrapping in a language-learning environment. *Journal of Computer Assisted Learning*, 19(4):90–102, 2003.

J.-C. Wu, Y.-C. Chang, T. Mitamura, and J. S. Chang. Automatic collocation suggestion in academic writing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010): Short Papers*, pages 115–119, 2010.

H. Yannakoudakis and T. Briscoe. Modeling coherence in ESOL learner texts. In *Proceedings of the 7th Workshop on Building Educational Applications Using NLP (BEA 2012)*, pages 33–43, 2012.

H. Yannakoudakis, T. Briscoe, and B. Medlock. A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*, volume 1, pages 180–189, 2011.

X. Yi, J. Gao, and W. B. Dolan. A Web-based English Proofing System for English as a Second Language Users. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pages 619–624, 2008.

Z. Yuan and M. Felice. Constrained grammatical error correction using Statistical Machine Translation. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL 2013): Shared Task*, pages 52–61, 2013.

Appendices

Appendix A

Metadata statistics

Table A.1: L1s represented in the data.

Code	Language	100 ANs	All ANs	100 VOs	All VOs
ak	Akan	0	2	0	1
ar	Arabic	0	12	2	15
as	Assamese	0	0	0	1
bg	Bulgarian	0	1	2	4
bn	Bengali	0	2	0	0
ca	Catalan	2	21	4	21
cs	Czech	1	18	2	12
da	Danish	1	4	2	5
de	German	14	90	11	76
el	Greek	11	111	16	81
en	English	1	8	0	6
es	Spanish	13	109	15	134
et	Estonian	0	1	0	0
eu	Basque	0	1	0	1
fa	Persian	0	5	0	6
ff	Fulani	0	2	0	0
fr	French	13	74	10	85
gsw	Swiss German	1	24	3	25
gu	Gujarati	0	1	0	2
hi	Hindi	2	14	0	10
hr	Croatian	0	0	0	2
hu	Hungarian	0	5	0	6
id	Indonesian	0	2	0	1
it	Italian	3	45	5	42
ja	Japanese	2	16	2	21
km	Khmer	1	1	0	0
kn	Kannada	0	0	0	1
ko	Korean	3	13	0	16
ku	Kurdish	0	1	0	0

Continued on next page

Table A.1 – *Continued from previous page*

Code	**Language**	**100 ANs**	**All ANs**	**100 VOs**	**All VOs**
lv	Latvian	0	0	0	1
men	Mende	0	2	0	0
ml	Malayalam	0	2	0	2
mr	Marathi	0	2	0	0
ms	Malay	0	0	0	1
my	Burmese	1	1	0	0
nl	Dutch	1	14	2	10
no	Norwegian	0	0	0	1
pa	Panjabi	0	4	0	1
pl	Polish	15	58	6	46
ps	Pashto	0	1	1	3
pt	Portuguese	13	61	5	61
rm	Romansch	0	0	0	1
ro	Romanian	1	9	1	11
ru	Russian	2	28	1	17
sd	Sindhi	0	2	0	1
si	Singhalese	1	4	0	3
sk	Slovak	1	6	0	0
sl	Slovene	0	2	0	2
sq	Albanian	0	0	0	1
sr	Serbian	2	4	0	0
sv	Swedish	5	26	3	12
ta	Tamil	0	2	4	14
te	Telugu	0	2	0	2
th	Thai	1	5	2	10
tl	Tagalog	0	4	0	5
tr	Turkish	0	9	3	17
ur	Urdu	0	7	2	6
vi	Vietnamese	0	0	0	2
yap	Yapese	0	0	0	1
zh	Chinese	17	103	11	95

CEFR Level	Exam	100 ANs	All ANs	100 VOs	All VOs
A1 (Basic User)	LNRSFLE1 (Skills for Life, Entry 1)	0	2	0	2
A1–A2 (Basic User)	KET (Key English Test)	6	49	8	55
A2 (Basic User)	LNRSFLE2 (Skills for Life, Entry 2)	3	10	0	3
A2–B1 (Basic to Independent User)	BEC1 (Business English Certificate 1)	0	2	2	6
	BECP (Business English Certificate – preliminary)	5	45	6	44
Basic (∼A)			**11%**		**12%**
B1 (Independent User)	PET (Preliminary English Test)	6	56	6	66
	LNRSFLE3 (Skills for Life, Entry 3)	2	6	0	1
	CELSP (Certificate in English Lang. Skills – Preliminary)	2	14	5	18
B1–C2 (Independent to Proficient User)	IELTSA (IELTS academic)	9	79	7	75
	IELTSG (IELTS general training)	4	28	5	32
B2 (Independent User)	FCE (First Certificate in English)	15	125	16	117
	BEC2 (Business English Certificate 2)	0	5	0	10
	BECV (Business English Certificate – vantage)	5	36	4	35
	LNRSFLL1 (Skills for Life, Level 1)	2	9	1	5
	CELSH (Certificate in English Lang. Skills – Higher)	2	14	0	4
B2–C1 (Independent to Proficient User)	ILEC (International Legal English Certificate)	3	8	2	8
	ICFE (Cambridge English: Financial)	1	2	0	1
Independent (∼B)			**40%**		**40%**
C1 (Proficient User)	CAE (Certificate in Advanced English)	17	158	18	154
	BEC3 (Business English Certificate 3)	4	16	2	14
	BECH (Business English Certificate – higher)	4	63	9	80
	LNRSFLL2 (Skills for Life, Level 2)	0	0	0	4
	CELSV (Certificate in English Lang. Skills – Vantage)	0	9	2	15
C2 (Proficient User)	CPE (Certificate of Proficiency in English)	40	216	25	177
Proficient (∼C)			**49%**		**48%**

Table A.2: Exam types.

Year	100 ANs	All ANs	100 VOs	All VOs
1993	31	206	23	207
1997	3	18	0	11
1998	10	46	4	38
1999	2	26	3	31
2000	4	33	4	41
2001	6	49	10	43
2002	4	55	7	62
2003	5	82	10	78
2004	7	56	4	40
2005	14	82	10	83
2006	5	50	12	76
2007	10	73	6	60
2008	27	173	25	166
2009	1	2	0	0

Table A.3: Examination years represented in the data.

Appendix B

Content word combination datasets

Tables B.1 and B.2 present the datasets in more detail:

1. Column 1 contains the identifier for the adjective/verb.
2. Column 2 contains the adjective/verb.
3. Column 3 contains the estimation of the frequency of the combinations with the adjective/verb in the BNC. The first figure represents the total number of occurrences (the number of *tokens*) of the ANs with the given adjective or the VOs with the given verb in the BNC. The figure in brackets represents the number of *types* – the count of the *unique* ANs with the given adjective or VOs with the given verb. These counts show how productive the adjective/verb is in native data.
4. Column 4 shows the number of examples with the given adjective/verb in the datasets.
5. Column 5 presents the error rate for the given adjective/verb calculated on the annotated CLC FCE dataset. It is estimated as the proportion of ANs/VOs with the given adjective/verb annotated with the codes `RJ`/`RV`. These error rates were used to choose the most problematic adjectives/verbs for language learners.
6. Column 6 presents the error rate for the given adjective/verb calculated on the dataset using OOC annotation. It is estimated as the proportion of ANs/VOs with the given adjective/verb annotated with the error code `I` out-of-context, additionally checking that the combination is incorrect due to the incorrect use of the adjective/verb (the error code on adjective/verb). The combinations annotated as incorrect due to the incorrect use of nouns do not contribute to calculation of the error rate. Error rates higher than 0.50 are marked in bold.
7. Column 7 presents the error rate for the given adjective/verb calculated on the dataset using IC annotation. It is estimated as the proportion of ANs/VOs with the given adjective/verb annotated with the error code `C` or `I` out-of-context, and further annotated as incorrect in context due to the incorrect use of the adjective/verb (the error code on adjective/verb). The combinations annotated as incorrect due to the

incorrect use of nouns do not contribute to calculation of the error rate. Error rates higher than 0.50 are marked in bold. The error rates for in-context annotation are always higher or equal to those for out-of-context annotation.

8. Column 8 presents the error codes used on the adjectives/verbs in the ANs/VOs annotated as incorrect, with the list of corrections suggested by the annotators.

Table B.1: AN combinations.

N	**Adjective**	**Frequency (BNC)**	**Frequency (dataset)**	**Error rate (CLC)**	**Error rate (OOC)**	**Error rate (IC)**	**Confusion set**
1	actual	5798 (1635)	6	0.590	0.00	**0.50**	S: *current*, *existing*
2	ancient	3922 (1293)	8	0.018	0.00	0.125	S: *old*
3	appropriate	5964 (1341)	7	0.154	0.00	0.14	S: *suitable*
4	bad	7338 (1096)	29	0.005	0.03	0.10	S: *inappropriate*, *poor* N: *big*, *serious*
5	best	16181 (2413)	42	0.001	0.00	0.12	S: *favourite*, *highest* N: *major*, *most* + adj
6	big	18157 (3089)	54	0.040	0.20	0.22	S: *broad*, *extensive*, *great*, *heavy*, *high*, *large*, *long*, *serious*, *strong*
7	bigger	1496 (671)	15	0.108	0.40	0.47	S: *greater*, *larger*, *longer*, *wider*
8	biggest	3938 (1169)	15	0.024	0.00	0.07	S: *greatest*
9	certain	16279 (2643)	13	0.045	0.08	**0.54**	S: *detailed*, *several*, *some*, *specific*, *various*
10	classic	2139 (872)	3	0.313	0.00	**1.00**	F: *classical*; S: *typical*
11	classical	2655 (738)	5	0.063	0.20	**0.60**	F: *classic*; S: *ordinary*, *standard*
12	clear	6645 (1190)	6	0.058	0.00	0.33	F: *clean*; N: *clever*
13	common	11893 (1626)	16	0.363	0.00	**0.56**	S: *joint*, *ordinary*, *standard*, *team*, *typical*; N: *political*
14	convenient	798 (343)	7	0.176	0.00	0.14	S: *suitable*
15	correct	3027 (768)	6	0.088	0.00	0.17	S: *wrong*
16	deep	5997 (1336)	6	0.027	0.00	0.33	N: *best*, *great*, *kind*, *solemn*
17	different	33964 (2827)	33	0.003	0.00	0.15	S: *distinctive*, *other*, *various*
18	economic	17165 (1439)	7	0.143	0.00	0.29	S: *financial*, *profitable*
19	economical	164 (101)	25	0.750	0.48	**0.92**	F: *economic* D: *economy* N: *rich*
20	elder	841 (129)	4	0.391	0.25	**1.00**	F: *elderly*, *older*
21	electric	2801 (583)	5	0.185	0.20	**0.80**	F: *electrical*, *electronic*; N: *credit*
22	electrical	1816 (360)	4	0.192	0.25	0.25	F: *electric*
23	false	2366 (564)	2	0.172	0.00	**0.50**	S: *wrong*
24	far	3894 (681)	4	0.095	0.25	0.25	S: *distant*
25	fast	1787 (616)	11	0.044	0.09	0.09	S: *fast-changing*, *fast-moving*
26	full	18113 (2302)	9	0.033	0.22	0.44	S: *complete*, *comprehensive*, *deep*, *great*, *total*, *profound*
27	funny	1435 (387)	14	0.090	0.00	0.21	F: *fun*; S: *strange*
28	further	18881 (2813)	10	0.010	0.10	0.30	S: *additional*, *other*
29	good	47937 (4206)	56	0.004	0.00	0.02	S: *big*
30	great	34470 (4691)	33	0.007	0.06	0.24	S: *big*, *good*, *high*, *huge*, *large*, *marked*, *sharp*, *significant*, *small*, *strong*, *substantial*, *tiny*

Continued on next page

Table B.1 – *Continued from previous page*

N	Adjective	Frequency (BNC)	Frequency (dataset)	Error rate (CLC)	Error rate (OOC)	Error rate (IC)	Confusion set
31	greatest	4248 (1156)	9	0.015	0.11	0.44	S: *best*, *biggest*, *highest*
32	hard	7228 (1172)	9	0.040	0.00	0.22	F: *hard-working*; S: *strong*
33	heavy	6662 (1575)	12	0.043	0.33	0.33	N: *serious*, *sharp*, *steep*, *strong*
34	high	22199 (2611)	5	0.017	**0.60**	**0.80**	S: *great*, *large*
35	historical	4436 (874)	13	0.268	0.23	0.31	F: *historic*, *history*; N: *traditional*
36	important	18073 (1860)	32	0.015	0.06	0.44	S: *large*, *major*, *serious*, *sharp*, *significant*; N: *big*, *famous*, *well-known*
37	incorrect	296 (185)	2	0.056	0.00	0.00	–
38	large	26736 (3382)	9	0.042	0.33	**0.55**	S: *baggy*, *big*, *broad*, *long*, *significant*, *strong*, *substantial*
39	latest	5262 (1387)	6	0.022	0.00	0.00	–
40	magic	1018 (331)	1	0.273	0.00	0.00	–
41	main	19245 (2020)	10	0.008	0.00	0.20	S: *most important*
42	near	1027 (229)	3	0.116	**1.00**	**1.00**	F: *nearby*
43	nearest	1610 (570)	4	0.056	0.00	0.25	S: *immediate*
44	nice	4483 (1233)	54	0.007	0.09	0.18	S: *best*, *good*, *kind*, *lovely*
45	particular	20424 (2730)	7	0.053	0.00	0.43	S: *special* N: *characteristic*, *typical*
46	precious	1119 (457)	5	0.102	0.20	**0.60**	S: *great*, *luxurious*, *valuable*
47	present	9876 (1746)	4	0.063	0.00	0.00	–
48	proper	4845 (1386)	14	0.099	0.00	0.43	S: *appropriate*, *respectable*, *right*, *suitable*; N: *individual*, *genuine*, *normal*, *own*, *real*, *regular*
49	rapid	2752 (663)	1	0.667	0.00	0.00	–
50	short	10679 (1441)	13	0.014	0.08	0.23	S: *brief*, *slight*
51	small	33279 (4224)	21	0.012	0.48	**0.52**	S: *brief*, *light*, *minor*, *narrow*, *restricted*, *short*, *tight*
52	soft	3719 (1122)	5	0.073	0.00	0.20	N: *light*
53	strong	10189 (1763)	19	0.088	**0.63**	**0.63**	S: *big*, *extreme*, *fierce*, *great*, *heavy*, *high*, *intense*, *loud*, *profound*, *serious*, *severe*, *strict*; N: *considerable*, *deep*, *extensive*
54	suitable	2911 (1033)	8	0.034	0.00	0.00	–
55	true	5366 (1771)	6	0.074	0.00	0.33	S: *genuine*, *real*
56	typical	2898 (1281)	18	0.135	0.00	0.44	S: *authentic*, *familiar*, *local*, *traditional*
57	unique	2574 (865)	8	0.143	0.00	0.25	S: *exclusive*, *only*
58	usual	4350 (1614)	4	0.114	0.00	0.00	–
59	various	13641 (2357)	10	0.069	0.10	0.30	F: *varied*; S: *different*
60	whole	20141 (2222)	15	0.008	0.00	0.13	S: *complete*, *full*; N: *real*
61	wrong	3806 (771)	12	0.032	0.08	**0.50**	S: *bad*, *false*, *inaccurate*, *mistaken*, *unfounded*, *unsuitable*

Table B.2: VO combinations.

N	Verb	Frequency (BNC)	Frequency (dataset)	Error rate (CLC)	Error rate (OOC)	Error rate (IC)	Confusion set
1	achieve	6039 (1170)	8	0.103	0.375	**0.625**	S: *reach*; N: *conduct*, *fulfil*, *meet*
2	acquire	2849 (934)	5	0.107	0.40	**0.80**	S: *achieve*, *get*
3	adapt	509 (277)	6	0.116	0.00	**0.50**	M: *adapt to*; F: *adopt*; N: *apply*
4	adopt	3348 (729)	8	0.134	0.375	**0.625**	F: *adapt to*; N: *gain*, *have*, *hire place*
5	affect	5903 (1568)	5	0.039	0.00	0.00	–
6	answer	3095 (401)	6	0.021	**0.50**	**0.50**	S: *reply to*, *respond to*; N: *do*, *sit*
7	ask	11755 (2942)	23	0.011	**0.61**	**0.70**	M: *ask about / for*; N: *order*
8	assure	543 (337)	4	0.055	0.00	**0.50**	S: *guarantee*
9	attain	507 (217)	3	0.073	**0.67**	**1.00**	F: *attend*, *maintain*
10	attend	4170 (627)	20	0.059	**0.65**	**0.65**	M: *attend to*; F: *attain*; S: *go to*, *participate in*; N: *expect*, *use*
11	avoid	5857 (1911)	14	0.043	0.07	0.29	N: *prevent*
12	bare	102 (38)	2	0.091	**1.00**	**1.00**	F: *bear*
13	bear	4307 (1043)	6	0.083	0.00	0.00	–
14	bring	15386 (4071)	19	0.065	0.00	0.16	F: *bought (∼brought)*; N: *work*
15	care	393 (210)	5	0.095	**1.00**	**1.00**	M: *care about*; S: *look after*
16	catch	4954 (1227)	8	0.105	0.125	**0.875**	S: *get*, *grab*; N: *collect*, *take*
17	close	4223 (693)	7	0.028	0.13	0.13	N: *switch off*
18	concern	3723 (1562)	9	0.014	0.00	0.11	N: *engage*, *hire*
19	contain	10276 (2836)	10	0.183	0.00	0.40	S: *have*, *hold*, *include*; N: *run*
20	cook	584 (188)	5	0.037	0.20	0.20	S: *bake*
21	cross	3358 (727)	5	0.066	**0.60**	**0.80**	S: *go through*; N: *go over*, *fly over*
22	describe	6819 (2295)	12	0.022	0.00	0.08	S: *express*
23	do	36998 (6124)	12	0.102	0.25	0.25	N: *exact*, *get*, *give*, *play*, *take*
24	earn	2332 (630)	5	0.018	**0.60**	**0.80**	S: *acquire*, *get*; N: *generate*, *meet*, *satisfy*
25	effect	488 (266)	11	0.750	**0.82**	**1.00**	F: *affect*; N: *experience*
26	enjoy	6147 (1540)	27	0.006	0.00	0.07	S: *benefit from*, *like*
27	ensure	3283 (1067)	4	0.034	0.00	**0.50**	M: *ensure for* S: *confirm*
28	experience	2316 (881)	8	0.013	0.00	0.00	–
29	fall	1900 (620)	5	0.333	**0.60**	**0.60**	F: *feel* N: *have*
30	feel	9059 (1955)	14	0.033	0.14	0.21	N: *see*, *smell*
31	gain	4894 (867)	11	0.135	0.27	0.45	S: *get*, *obtain*, *win*; N: *experience*, *reach*
32	get	44905 (6238)	18	0.056	0.22	0.39	D: *get+noun −> verb*; M: *get to*; S: *achieve*, *obtain*, *receive*; N: *give*, *have*, *take*
33	hope	460 (343)	4	0.056	0.25	**0.75**	M: *hope for*; N: *look forward to*, *wait for*
34	hurt	577 (249)	4	0.114	0.25	**0.50**	S: *damage*, *harm*
35	imply	1594 (747)	8	0.089	0.25	**0.50**	S: *call for*, *entail*; N: *cause*

Continued on next page

Table B.2 – *Continued from previous page*

N	Verb	Frequency (BNC)	Frequency (dataset)	Error rate (CLC)	Error rate (OOC)	Error rate (IC)	Confusion set
36	increase	11060 (1563)	15	0.044	0.40	0.40	S: *expand*, *extend*; N: *foster*, *improve*
37	inform	1424 (670)	6	0.111	**0.50**	**0.50**	M: *inform of* N: *spread*, *tell*, *give*
38	join	8000 (2318)	18	0.049	**0.72**	**0.83**	S: *combine*; N: *attend*, *become*, *enlist in*, *enter*, *experience*, *follow*, *get into*, *have*, *pursue*, *share*
39	know	14905 (3770)	16	0.027	0.06	0.25	S: *experience*, *get to know*, *visit*; N: *find out about*
40	lead	7440 (2081)	6	0.103	0.33	0.33	M: *lead to*; N: *live*
41	learn	4116 (808)	16	0.046	0.31	0.31	M: *learn about*; S: *acquire*
42	live	3467 (836)	7	0.082	**0.86**	**0.86**	M: *live in / through*; F: *leave*, *relive*; S: *experience*; N: *have*, *share*
43	look	4473 (1528)	9	0.042	**1.00**	**1.00**	M: *look at*; F: *took*; S: *watch*; N: *make*
44	loose	65 (59)	33	0.911	**1.00**	**1.00**	F: *lose*
45	lose	10762 (1958)	20	0.040	0.05	0.20	N: *leave*, *miss*
46	make	85295 (7342)	22	0.130	0.45	**0.55**	D: *make+noun –> verb*; S: *create*, *set (up)*; N: *become*, *do*, *have*, *hold*, *take*, *sit*
47	obtain	5077 (1213)	10	0.100	**0.60**	**0.80**	S: *achieve*, *get*, *take*; N: *buy*, *make*, *receive*, *win*
48	open	9688 (1303)	9	0.021	0.22	0.22	F: *open up*; N: *turn on*
49	pay	12514 (1637)	15	0.009	**0.80**	**0.80**	M: *pay for*; N: *buy*, *make*, *offer*
50	place	5041 (1650)	7	0.089	0.29	**0.57**	S: *locate*, *put*; N: *conduct*, *make*
51	play	14264 (2305)	7	0.008	0.29	0.29	N: *perform*, *put on*
52	prepare	2854 (875)	29	0.016	0.28	0.38	M: *prepare for / to*; S: *arrange*, *make*; N: *provide*, *save up*
53	prevent	6219 (2349)	6	0.072	0.17	**0.50**	S: *block*; N: *protect*, *warn*
54	propose	1894 (669)	12	0.448	0.00	**0.50**	S: *offer*, *suggest*; N: *charge*
55	put	25542 (4412)	9	0.079	0.22	0.44	S: *install*; N: *add*, *impose*, *include*, *introduce*
56	raise	10932 (1522)	3	0.129	0.33	**0.67**	S: *improve*, *increase*, *enhance*; N: *call*, *stage*
57	reach	10614 (2477)	26	0.053	0.42	0.42	N: *achieve*, *attain*, *grasp*, *have*, *receive*, *understand*
58	remind	835 (412)	5	0.550	**0.60**	**0.80**	S: *recall*, *remember*; N: *amend*
59	request	890 (450)	13	0.019	0.00	0.08	S: *ask for*
60	rise	1450 (547)	11	0.615	**0.82**	**0.82**	F: *raise*; S: *increase*; N: *draw*
61	save	4165 (1288)	5	0.014	0.20	0.20	N: *reduce*
62	solve	1872 (188)	9	0.027	0.56	0.56	S: *tackle*; N: *deal with*, *ease*,

Continued on next page

Table B.2 – *Continued from previous page*

N	Verb	Frequency (BNC)	Frequency (dataset)	Error rate (CLC)	Error rate (OOC)	Error rate (IC)	Confusion set
							eliminate, *manage*, *save*
63	speak	1700 (458)	5	0.022	0.40	0.40	M: *speak with*; N: *spend*
64	spend	8898 (857)	10	0.006	0.30	0.40	M: *spend on*; S: *pay*; N: *hold*, *open*
65	stop	4765 (1880)	16	0.026	0.06	0.06	N: *put down*
66	suffer	3352 (703)	5	0.020	0.40	0.40	S: *endure*; N: *reach*
67	suggest	3885 (1622)	16	0.046	0.00	0.25	S: *recommend (to)* N: *choose*
68	take	84449 (6636)	19	0.039	0.16	0.21	D: *take+noun −> verb*; N: *get*
69	teach	2005 (696)	1	0.031	0.00	0.00	–
70	tell	14354 (3231)	17	0.049	0.12	0.24	S: *recite*, *recount*, *say*; N: *give*, *mention*
71	wait	1257 (258)	12	0.046	**0.92**	**0.92**	M: *wait for*; S: *expect*, *hope for*, *look forward to*
72	want	12829 (3614)	10	0.013	0.10	0.10	N: *have*
73	watch	6565 (2187)	9	0.026	0.00	0.33	S: *look at / upon*, *see* N: *look after*
74	wear	6339 (1073)	8	0.035	0.00	0.00	–
75	win	10394 (1473)	4	0.034	**0.50**	**0.50**	N: *beat*, *make*
76	wish	542 (376)	6	0.049	**0.50**	**0.67**	M: *wish for* S: *want*