A Corpus-Based Collocation Assistant for Swedish Text

ROBERT ÖSTLING

KTH Computer Science
and Communication

Master of Science Thesis
Stockholm, Sweden 2009

A Corpus-Based Collocation Assistant for Swedish Text

ROBERT ÖSTLING

Master's Thesis in Computer Science (30 ECTS credits)
at the School of Computer Science and Engineering
Royal Institute of Technology year 2009
Supervisor at CSC was Ola Knutsson
Examiner was Stefan Arnborg

TRITA-CSC-E 2009:124
ISRN-KTH/CSC/E--09/124--SE
ISSN-1653-5715

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
SE-100 44 Stockholm, Sweden

URL: www.csc.kth.se

A Corpus-based Collocation Assistant for Swedish Text

Abstract

A *collocation* is a recurrent combination of words, such as *commit a crime*, whose meaning is fairly transparent but which can not be changed arbitrarily, even if the rules of grammar are adhered to and the individual meanings of the words are preserved. If a writer tries to use *do a crime*, the job of a *collocation assistant* is to suggest a better alternative—such as *commit a crime*—just like a spell checker would do for spelling errors.

While spell checkers and grammar checkers have become everyday tools, collocation assistants are still in the prototype stage. In the last couple of years, a series of articles have been published on the development of several different tools aimed at learners of English.

This report describes the design, implementation and evaluation of the first collocation assistant for written Swedish. Novel features include the possibility to use a Random Indexing vector space model for measuring semantic similarity, and the use of a full dependency parser.

Ett korpusbaserat kollokationsverktyg för svensk text

Sammanfattning

En *kollokation* är en återkommande ordkombination, exempelvis *begå brott*, vars mening är relativt uppenbar men som inte kan ändras godtyckligt, trots att alla grammatiska regler följs och de enskilda ordens betydelser bevaras. Om en skribent försöker använda *göra brott* är en *kollokationsassistents* uppgift att föreslå ett bättre alternativ, exempelvis *begå brott*, precis som en stavningskontroll gör för stavfel.

Medan stavnings- och grammatikkontroll har tagit steget in i vardagen, är kollokationsassistenter fortfarande i prototypstadiet. Under de senaste två åren har ett antal artiklar om verktyg för studenter som lär sig engelska publicerats.

Den här rapporten beskriver design, implementation och utvärdering av den första kollokationsassistenten för skriven svenska. Nyheter är att vektorrumsmodellen Random Indexing kan användas för att mäta semantisk likhet, och att en full dependensparser används.

Thanks

... to my supervisor Ola Knutsson, who contributed the original idea of this project, materials, advice, proofreading, and encouraging words along the way.

... to the two anonymous reviewers of the work-in-progress report of this project (Östling and Knutsson, 2009) for numerous useful suggestions about the project and interpretations of its results.

... to Joakim Nivre and Johan Hall, for helping me to adapt their tool, MaltParser, to my project.

... to the Human Language Technology group at KTH, for introducing me to the field of computational linguistics and for creating many of the tools and resources I have used in my project.

... to Stefan Arnborg, for showing me that statistics does not have to be mind-numbingly dull. And for agreeing to examine this thesis.

... to everyone else who contributed suggestions and corrections.

Contents

1 Introduction

Spell checkers have become an essential part of word processors, cell phones and almost any computerized system dealing with text input in some way. While there are many details to pay attention to, the basic idea behind a spell checker such as CSC's Stava (Kann et al., 1998) is to check each word in a text against a list of accepted spellings. This process is efficient, reliable and widely implemented.

Grammar checkers are less ubiquitous than spell checkers, but have entered the ordinary computer user's home through modern word processing software. Different algorithms may be used. CSC's Granska (Domeij et al., 1999) uses a set of rules (for instance, to ensure that word inflections are consistent in a phrase) and warns the user if these rules are broken. While this is a more difficult task than spell checking, and the accuracy is lower, current grammar checkers are sufficiently developed to be used in consumer applications.

However, spelling and grammar errors are not the only ones to occur in natural language texts. Consider the following sentence, with perfect spelling and grammar, where every word is used in its normal sense, and whose meaning is clear enough:

> I start every day with a cup of powerful tea.

Something is not quite right here. We do not usually say *powerful tea*, but rather *strong tea*. This is often referred to as a *collocation error*, because intuitively speaking, there is a "better way of putting it." These errors are often difficult to detect and correct, even for humans.

In recent years there has been an increasing amount of research carried out attempting to create *collocation assistants*, automated tools that compare the input of the user with a previously collected text corpus, in order to suggest improvements to such awkward phrases as the one above.

The primary goal of this project is to create and evaluate a tool to find and correct collocation errors in a Swedish text. The purpose of such a tool is twofold: to detect collocation errors, primarily in texts written by non-native speakers of Swedish, and to suggest changes to texts by proficient writers, for the sake of variety or clarity.

A secondary goal of the project is to evaluate several different methods that have been used or suggested in the literature, even though not all of them will be put to practical use.

2 Theory

2.1 Collocations

The term *collocation* has been used in many different senses, but its basic meaning is a recurring pattern of words. In the widest sense, this could be extended to include patterns ranging from idioms to phrasal verbs, but it is often useful to limit the definition to a more specific class of patterns.

Nesselhauf (2005), Manning and Schütze (1999) and Malmgren (2002) survey the variety of definitions of the term "collocation" in the literature, in addition to providing their own definitions. This is a complex subject, and we can only summarize the main currents of previous research here.

2.1.1 Statistical definitions

The oldest and simplest class of definitions are statistical, due to J. R. Firth in the 1950s. A *collocation* is considered to be a pattern of words that occur together several times in a corpus that is representative of the language. "Together" can refer to any kind of relation between the words, from simple word pairs like *strong tea*, to more complex phrases like "X *shaped* Y's *understanding of* Z" involving three non-consecutive words. In a sufficiently large and varied corpus, we are likely to find several instances of the phrase *strong tea*, but, unless the corpus includes articles on linguistics, the seemingly similar phrase *powerful tea* is unlikely to occur. Without any knowledge of what *strong* and *tea* mean, or even which parts of speech they are, a computer program could discover that they occur together a statistically significant amount of times.

There are a couple of problems with this approach. First, no corpus is large enough to cover all possible combinations. Malmgren (2002) cites a study where *last April* but not *last August* was identified, even though these are obviously two instances of the same "last *month*" pattern. Second, there are many phrases that occur frequently in a corpus, even though the words don't "belong together" in an intuitive sense. For instance, we are likely to find many examples of the phrase *buy a book*, but this can be explained entirely by the basic semantic meanings of *to buy* and *a book*, and the common grammatical pattern *verb+noun*. This distinction is important theoretically, and touches on the difference between the two main schools of thought regarding collocations.

2.1.2 Constructionist definitions

The second group of definitions of the term *collocation* focuses on the role of collocation in a language, rather than simply looking at statistical co-occurrences in a corpus. Here it is useful to look at human language from a constructionist point of view (Goldberg, 2003). A *construction* is any unit in a language whose meaning or frequency can not be explained entirely through its components. At the lowest level we have morphemes such as the *pre-* prefix in *pre*fix. There is nothing in the sound *pre* that would make it obvious that its meaning is *before*, it is simply used as such. At the next level there are words, such as *prefix*. While in some cases morphemes indicate the meaning of a word, the precise meanings can often not be determined with certainty. For instance, one may "*put* family *before* work," but hardly "let family *prefix* work" even though *prefix* literally means *put before*. Thus *prefix* is a separate construction in English. At a higher level, there are the traditional sentence patterns of English grammar, such as "*subject* is *verb*ing *object*." An instance of this construction would be the phrase "*I* am *pull*ing *your leg*" in its literal sense. However, this phrase can also be a different type of construction, an idiomatic phrase, "*subject* is pulling *object*'s leg" with the meaning that the subject is joking with the object.

From this point of view, collocations are simply a type of constructions. There is a construction "strong *beverage*" meaning a beverage that has a high concentration of some substance. *Strong tea* and *strong coffee* are phrases using this construction, while e.g. *powerful tea* is not. *Buy*, *book* and "*buy*/*purchase* an *item*" could be seen as constructions, while the phrase *buy a book* is not a construction on its own, but simply a result of combining these other constructions. "*Buy an official*," on the other hand, is a construction with a meaning similar to "*bribe an official*."

Now, *why* would a language be constructed in this haphazard fashion, instead of the much more elegant generative model where we essentially are believed to have a memorized vocabulary and a mental "grammar unit" to compose and decompose sentences made of these vocabulary items? Constructionists argue that the human brain is simply better at memorization than it is at quickly creating and analyzing sentences, so for efficient communication we need these ready-made chunks of language. Shei (2005) discusses the view of language reuse across cultures, and Morgan Lewis echoes the constructionist view from the point of view of an English language teacher (Lewis, 2000, p. 15):

> No wonder students make so many grammar mistakes! They are using grammar to do what it was never meant to do. Grammar enables us to construct language when we are unable to find what we want ready-made in our mental lexicons.

If our goal is to write a tool dealing with collocations, this view of collocations has one major drawback: how, if statistical methods are such poor indicators, are we supposed to know what the collocations of a language are? Unfortunately, this is in the domain of human lexicographers, and a very time-consuming task. For this reason, collocation assistants generally do not pay much attention to the finer linguistic points, but focus on easy-to-measure statistical properties.

2.1.3 Lexical functions

Malmgren (2002) describes a theoretical framework that is useful for certain types of collocations.

A *lexical function* represents a semantic function in the language, such as "carrying out an action." This is illustrated in table 1. We may refer to this function as $\mathbf{Act}(x)$, which takes a noun describing an action, x, and returns a set of verbs that may be used in the sense of carrying out this action.

Malmgren (2002) quotes studies suggesting that there are about 50–60 such functions, and that this remains fairly constant across languages. However, the values of the functions do vary, even between related languages. This is a common source of miscollocations among non-native speakers.

Lui et al. (2008) use this idea in their collocation assistant tool (see section 2.6), by assuming that users are more likely to confuse words within the range of the same lexical function.

Table 1: Example of a lexical function that describes the carrying out of some action.

x	**Act**(x)
task	{ carry out, perform }
crime	{ carry out, commit }
marathon	{ complete, run }

Table 2: The four collocation types considered by Futagi et al. (2008).

Pattern	**Examples**
verb + noun (direct object)	*reject an appeal*
adjective/noun + noun	*strong tea*, *house arrest*
noun *of* noun	*swarm of bees*
verb + adverb	*argue strenuously*

2.1.4 Collocation types

The simplest and most often investigated types of collocations are those which can be expressed easily by lexical functions. There is one *basis*, and to express the semantic meaning represented by the lexical function **F**(x) (such as **Act**(x) described in the previous section), we can use the *collocates* in the set **F**(*basis*).

Common examples, here given with the basis in **bold**, include verb-**noun** collocations (*commit* ***crime***, *make* ***mistake***) and adjective-**noun** collocations (*stiff* ***breeze***, *strong* ***tea***).

The tools of Chang et al. (2008) and Lui et al. (2008) focus exclusively on verb-noun miscollocations, which are found to be a particular problem for non-native speakers of English. According to Futagi et al. (2008), 87% of miscollocations in a corpus of Taiwanese learners of English were found to be of the type verb-noun, with the verb as the basis and the noun as collocate in 96% of those cases.

Futagi et al. (2008) use four different collocation types, listed in table 2.

Finally, the tool of Park et al. (2008) does not classify collocations, but simply looks at word n-grams. One advantage of such a method is the ability to handle complex set phrases, such as *the proverbial needle in the haystack*.

2.1.5 Frequency of collocations

We should not *think of* collocations *as* some linguistic curiosity like proverbs, which are *far apart* and do not form an important part of the language. Rather, they are very frequent and *essential to* the language.

Nesselhauf (2005) studied verb-noun collocations in a corpus of 154,191 words *consisting of* essays by *native* German *speakers* with *at least* one *year of* English studies at university level *behind them*, and found 2,082 such verb-noun collocations. That is *one every* 74 words, or assuming an average *sentence length* of 15–20 words, approximately *one every* four sentences of *this specific* collocation type.

I have put recurring multi-word expressions in the previous paragraph in cursive, to demonstrate how unexpectedly frequent these constructions are, which can reasonably be considered collocations according to some of the definitions discussed earlier.

2.2 Non-collocations

Collocations are important, but what we are really interested in are phrases that are *not* collocations, because some of these may need to be corrected.

2.2.1 Miscollocations

The goal of writers, we may assume, is to express their thoughts in a way that a skilled native speaker would. This includes getting collocations right. For instance, the Swedish phrase *fatta eld* (literally *grab fire*) is properly translated as *catch fire.* At a glance, both translations would seem about equally reasonable, but only *catch fire* is a construction in English describing something where combustion is beginning, so if *grab fire* were to be used in this sense we would label it a *miscollocation.*

Grab fire happens to be very uncommon in English, so a human or a computer program with access to an English corpus could easily guess that another phrase was intended. With some context and a bit of imagination or a Swedish-English dictionary, we could probably guess that *catch fire* was the intended phrase. Cases like these, where the given phrase is very uncommon and there is a similar phrase that is very common, are where current collocation assistants excel.

However, there are more difficult cases. A writer whose native language is Chinese might try to translate 打電話 dǎ diànhuà (*make a phone call*) literally into *hit the telephone.* This is a grammatical English phrase, which is sometimes used in its literal sense of physical violence against a telephone. It would be difficult for a computer program to determine if this interpretation makes sense in context. But can we even say that this is an error? English does have phrases like *hit the sack* and *hit the road*, so the interpretation of *hit the telephone* as roughly synonymous to *make a phone call* should be clear.[1]

The lesson to be learned is that the author of a collocation assistant should be careful about choosing its goals. With spelling and grammar checkers, one simply wants to detect erroneous use of language (according to some authority), and perhaps suggest some correct alternatives. But how do we determine whether or not a collocation is "correct"? And if we can not tell right from wrong, how could we expect machines to do so?

Instead of *collocation errors*, phrases such as *non-native-like collocations* are often used, especially in the typical situation of second language learning. In general, collocation usage of native speakers is better than that of non-native speakers, but native speakers also confuse collocations. Take for instance the English phrase "I could care less" used in the sense of "I *couldn't* care less." It is unlikely that a non-native speaker would produce the rather nonsensical first version.

[1] An Internet search using Google's search engine (`http://www.google.com/`) reveals that this usage is rare, but does occur in at least one published work (Ingrid Seward: *The Queen and Di*)

Here I will use the term *miscollocation* to mean a phrase which may or may not have the intended meaning, but where there is another common phrase that a skilled user of the language would prefer to express the meaning intended.

Nesselhauf (2005, pp. 49–54) discusses the practical aspects of judging the acceptability of collocations. She uses a five-level scale based on the combined judgement of one or more speakers.

2.2.2 Non-collocations

We have discussed collocations and miscollocations, but there is another important category of phrases: non-collocations.

A collocation is a construction that is often used to express a certain meaning, usually adding some meaning beyond the words it is made up of.
Example: *strong tea.*

A miscollocation is a phrase that is not a collocation, but that is attempting to express something that there already is a collocation for, which makes the miscollocation appear clumsy at best, or even plain wrong.
Example: *powerful tea.*

A non-collocation is not a collocation, but the result of combining constructions to express something for which there is no collocation.
Example: *flammable tea.*

These distinctions are critical, since a collocation assistant should accept collocations, correct miscollocations, and ignore non-collocations.

2.3 Statistical collocation extraction

As we have seen, the task of finding collocations in a language is very difficult, even for humans. Computers are currently unable to solve this general task, so one normally focuses on the more specific problem of finding recurring phrases in a text corpus.

There has been much research into various methods for statistical collocation extraction (Evert, 2005; Cinková et al., 2006; Johansson, 2001; Krenn and Evert, 2001; Lin, 1998; Manning and Schütze, 1999; Pearce, 2002; Petrović et al., 2006; Quasthoff and Wolff, 2002; Smadja, 1993).

2.3.1 Model

While there are generalizations, we will only consider word *pairs*, since both of the collocation types our tool is to handle can be modelled this way.

The goal is to find out how strongly two words are associated. We start with formulating a null hypothesis, where words are assumed to occur with different frequencies and independent of each other.

Table 3 is the *contingency table* of the observed data (Evert, 2005). O_{11} is the number of co-occurrences of u and v, O_{22} is the number of pairs with neither u nor v, O_{12} is the number of times that u occurs without being paired with v, and vice versa for O_{21}. R_i is the sum of row i, that is, R_1 is the total number of occurrences of u and $R_2 = N - R_1$ where N is the total number of words. C_i is the corresponding data for the columns, that is, word v.

Given this information, and the assumption of independence, we can compute the expected frequencies according to table 4.

Table 3: Observed frequencies, with respect to the word pair (u, v).

	$V = v$	$V \neq v$	Σ
$U = u$	O_{11}	O_{12}	R_1
$U \neq u$	O_{21}	O_{22}	R_2
Σ	C_1	C_2	N

Table 4: Expected frequencies, with respect to the word pair (u, v).

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

Evert (2005) discusses a wide range of *association measures* based on these tables. Here we will focus on three of the most used and useful measures: frequency, Mutual Information and log-likelihood.

2.3.2 Frequency

The frequency of a word pair, O_{11} in table 3, is a rough but often useful estimate of association strength. Unless the words u or v are very rare, there will be many examples present of a strong collocation in a large corpus. In fact, Park et al. (2008) found that users of their collocation assistant had trouble understanding more advanced association measures and preferred a simple frequency count.

The major downside of using only the frequency of the word pair, is that a fairly large count can be explained by u and v co-occurring randomly if they are common and the corpus is large. Conversely, if they are rare, the frequency will be low even though it may be higher than expected by a statistically significant amount.

2.3.3 Mutual Information

Mutual Information indicates how much more common than expected (by the null hypothesis) the observed frequency of the word pair is:

$$MI = \log \frac{O_{11}}{E_{11}}$$

Where O_{11} and E_{11} are defined as in tables 3 and 4. A negative value indicates that the word pair is *less* common than would be expected by chance, while a positive value means that it is *more* common. This makes Mutual Information a *one-sided association measure*, see Evert (2005, p. 75) for a more thorough discussion.

Another property of the Mutual Information value can be observed directly from its definition above: it depends only on the quotient O_{11}/E_{11}, not on the absolute number of observed instances. When O_{11} is small (as is often the case, even with a large corpus) the Mutual Information value is unstable. Evert

Table 5: Verb collocates of the noun *blomma* (flower).

Word (lemma)	N	MI	LL
ha (have)	178	-2.3610	1690.0597
lägga (lay, put)	36	-0.9808	25.0953
få (get)	29	-3.0454	529.5158
plocka (pick)	21	0.8778	6.1978
bära (carry, wear)	18	-0.5425	3.2229
ge (give)	16	-2.5425	150.5009
skicka (send)	15	-0.4566	1.8423
se (see)	15	-2.7960	194.4754
vattna (water)	13	3.4960	33.3064

(2005, p. 89) discusses different improvements to the basic Mutual Information definition, while Pearce (2002) only lets the value be defined for $O_{11} > 5$.

2.3.4 log-likelihood

The log-likelihood association measure indicates the probability that the frequency of co-occurrence is not due to chance, and is defined as:

$$LL = 2\Sigma_{ij} O_{ij} \log \frac{O_{ij}}{E_{ij}}$$

Evert (2005, p. 83) discusses this in some detail. We will simply note one important detail: log-likelihood is a *two-sided association measure*, meaning that a high value can indicate that the observed co-occurrence frequency is either significantly higher or significantly *lower* than expected. The latter occurs for non-collocations of two common words, since many co-occurrences would be expected in a large corpus if the distributions of the words were random.

2.3.5 Comparison of association measures

We finish our discussion on association measures by giving some examples, comparing the number of occurrences (N), Mutual Information score (MI) and log-likelihood score (LL). The statistics are taken from our corpus (see section 3.2).

Table 5 lists the most common verbs used with *blomma* (flower) as their direct object. As we can see, only two of these have positive Mutual Information scores, indicating that the frequency is higher than expected. Both of these have low log-likelihood scores and by absolute terms are only about 10% as common as the most common verb involving flowers: *to have*.

If we look at the adjective collocates to the noun *brott* (crime), the results look somewhat different (table 6).

Here the two most common adjectives have high Mutual Information *and* high log-likelihood scores, and are in fact near-synonymous in this sense, belonging to a lexical function magnifying the scope of the basis (in this case, *crime*). Further down we can see the non-collocation (as indicated by its very low Mutual Information score and high log-likelihood score) of *stor* (big). By

Table 6: Adjective collocates of the noun *brott* (crime).

Word	**N**	**MI**	**LL**
grov (serious)	217	3.2048	495.1850
allvarlig (serious)	160	2.6661	281.2083
olik (different)	44	-1.2014	50.1972
ny (new)	43	-1.7444	130.9826
misstänkt (suspected)	40	1.9615	44.3792
stor (big, great)	35	-2.5795	349.1649
anmäld (reported)	34	3.4669	86.0697

all logic, this should belong to the same set as the first two adjectives, but as a peculiarity of the Swedish language, it simply isn't.

The product of the Mutal Information and log-likelihood scores has the desirable property of taking both statistical significance and positive association strength into account. In section 3.5 this product is applied when looking for alternatives to miscollocations.

2.4 Synonyms

Collocation assistants (see section 2.6) typically make use of "synonyms" in order to generate and investigate alternatives to a phrase.

"Synonym" is actually not a good term for our purposes, since the words we are looking for are often not particularly close in meaning. For instance, we would like to guess that in the phrase "eventually the barn *got fire* and burned to the ground," the "synonym" of *get* that we are looking for is in fact *catch*, in spite of the basic meanings of the two verbs being quite distant. Both relate to obtaining something, but semantically they are too far apart to be listed in a synonym dictionary. There are even more extreme examples, such as *ta självmord* (literally *take suicide*) and *begå självmord* (*commit suicide*), which in modern informal usage are equivalent. But *take* and *commit* could hardly be viewed as synonyms in any traditional sense.

Nevertheless, collocation assistants commonly use synonym dictionaries or similar semantic resources to evaluate candidate expressions. For instance, Futagi et al. (2008), Lui et al. (2008) and Park et al. (2008) use WordNet (Fellbaum, 1998).

2.4.1 Folkets synonymlexikon

Folkets synonymlexikon (Kann and Rosell, 2005), literally *The People's Synonym Dictionary*, was initially constructed by automatic means from bilingual dictionaries and Random Indexing (see section 2.4.2). It is continually improved by anonymous users via the Internet, who can suggest new synonym pairs and vote on the quality of existing pairs in the database.

The result is fairly similar to a traditional synonym dictionary, except that each pair of synonyms also has a numeric value of how close in meaning the words are, according to the voters.

Table 7: Swedish "synonyms" from Folkets synonymlexikon (FSL) and our Random Indexing model (RI).

Word	**Source**	**Suggestions**
grå	FSL	dyster, tråkig
(*grey*)	FSL	(sullen, boring)
grå	RI	violett, ytlig, slug, lekfull, sträv
(*grey*)	RI	(violet, superficial, sly, playful, rough)
visa	FSL	lotsa, påvisa, synliggöra, exponera, indikera
(*show*)	FSL	(guide, demonstrate, display, expose, indicate)
visa	RI	signalera, påstå, besluta, markera, föreställa
(*show*)	RI	(signal, claim, decide, mark, introduce)

Unfortunately, while most synonym pairs are good, there are many spurious non-synonyms as well. There is also no separation of different parts of speech. For instance, Swedish adverbs and neuter-inflected adjectives usually look the same, and they are frequently confused in Folkets synonymlexikon.

2.4.2 Random Indexing

Random Indexing (Sahlgren, 2005) is an efficient vector space model which can be used to estimate the similarity of the contexts in which a certain word occurs. Words that occur in similar contexts often have related meanings.

The basic idea behind Random Indexing is that each *context* (such as neighboring words, phrases, part of speech tags, etc.) occurring in the corpus is assigned an *index vector* of several hundred to a few thousand elements, with almost all elements equal to zero, but a handful that are either 1 or -1. For each word we are interested in, we then add all the random vectors of its contexts, and obtain a *context vector* for that word. The Random Indexing similarity between two words, then, is the similarity (measured for instance by the dot product of the normalized vectors) between the context vectors of the words in question.

When one makes a list of the handful of words closest to any given word according to a Random Indexing model, the result is typically rather different from a classical synonym dictionary. Table 7 consists of some examples to illustrate this point, where the top five suggestions have been selected from the different sources.

The Random Indexing suggestions tend to have *some* relationship to the original word, even if it is rarely a "synonym." For instance, *grå* (*grey*) and *gul* (*yellow*) are both colors, *visa* (*show*, in the sense of prove) and *påstå* (*claim*) are used in similar contexts but have contrasting meanings, and *grå* (*grey*, in the sense of boring) and *playful* are antonyms. Such antonym pairs are rather common, and are particular concern since my experiments have shown that words often collocate with certain other words as well as the antonyms of these words.

For instance, if a Random Indexing model was to suggest that *strong* and *weak* are related (and presumed to be synonymous), then *strong tea* might very well be "corrected" into the opposite *weak tea*.

2.4.3 Bilingual dictionary

Chang et al. (2008) focus on second-language learners, and observe that many collocation errors are due to interference from the user's native language. They attempt to exploit this fact by using a bilingual dictionary (Chinese-English in their case) to translate a word back and forth, generating a set of words with the same translation in Chinese.

Chang et al. (2008) cite a study that found interference from Chinese in 84% of verb-noun miscollocations in a corpus of Chinese-speaking English learners, while Nesselhauf (2005) found interference from German in 51% of verb-noun miscollocations in a corpus of German-speaking English learners. It is difficult to determine with certainty what the cause of a particular miscollocation is, but on the basis of these numbers it is reasonable to expect that a bilingual dictionary can be useful.

This method requires a dictionary of the user's native language and the target language, something which is often not easily available, especially for a language like Swedish with relatively few speakers.

2.5 Text processing

A collocation assistant typically needs to process a corpus used for reference (since all tools that I am aware of are corpus-based) as well as the user's input. Sometimes there is considerable overlap in the processing performed.

2.5.1 Corpus

Many collocations are rare. For instance, *begå massaker* (*commit massacre*) occurs only five times in our corpus of 75 million words, on average once in 15 million words. Yet it is not an overly obscure phrase. This example should demonstrate that for statistical collocation analysis, we really do want a large corpus.

Chang et al. (2008) and Lui et al. (2008) use the British National Corpus (100 million words), Futagi et al. (2008) use one billion words compiled from different sources, Park et al. (2008) use the English Wikipedia[2] and US government web pages, of an unspecified total length. Guo and Zhang (2007) have not constructed a collocation assistant like the previously mentioned projects, but suggest using the Google[3] search engine, indexing many billion words of English text, for extracting collocations.

For a language with much fewer speakers, such as Swedish, the problem of finding a large enough corpus is even greater. There are two main corpora used for research in computational linguistics: The Stockholm Umeå Corpus, SUC[4] of 1 million words, and PAROLE[5] of 19 million words.

Not only research-oriented corpora have been used for collocation assistants, however. Park et al. (2008) use Wikipedia, whose Swedish edition[6] contains about 41 million words. Its quality varies but is generally sufficient for gathering

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

[3] http://www.google.com/

[4] http://www.ling.su.se/staff/sofia/suc/suc.html

[5] http://spraakbanken.gu.se/parole/

[6] http://sv.wikipedia.org/

statistics. One advantage that should not be overlooked of using Wikipedia as a text corpus is its liberal license, which permits redistribution.

2.5.2 Part of speech tagging

In some designs, it is necessary to know which part of speech (POS) the different words in a text are. For instance, Futagi et al. (2008) use a POS tagger before performing finite-state parsing (see section 3.3.4) in order to extract word pairs for analysis.

The state of art in POS tagging is fairly good, Carlberger and Kann (1999) report 97% accuracy in their tagger for Swedish, and it can be implemented efficiently. This makes POS tagging a minor issue in the context of a full collocation assistant.

2.5.3 Parsing

Some systems, like Park et al. (2008) simply use word n-grams directly from the input text, but this approach does not work when there is a large distance between the elements of a collocation. In the phrase "you have committed, and covered up, the most heinous of crimes!" we find the collocation *commit crime* with 7 words and two punctuation marks between *commit* and *crime*. It would be useful to know that *crime* is the object of the verb *commit*. This applies both to the user's text under analysis, and to the corpus where we gather our collocation statistics from.

Futagi et al. (2008) use finite-state parsing of the part of speech tagged text to find these relationships. This technique is also used in other applications, such as the rule-based grammar checker Granska (Domeij et al., 1999).

As far as I know, no previous collocation assistant uses a full parser. This is likely due to the fact that correctly parsing a text in English (or Swedish) is a very difficult task. MaltParser (Nivre et al., 2007) has a relatively low accuracy. 86.3% of words in a Swedish text are assigned the correct headword, 82.0% the right headword *and* the right relationship tag. The corresponding numbers for MaltParser on an English text are 88.1% and 86.3%. In addition, its performance is rather low (about 4 CPU hours per million words), so parsing a large corpus requires some resources.

2.5.4 Lemmatization

Lemmatization is the task of annotating each word in a text with its lemma form. For instance, the sentence "The cats were taught two lessons." has the following lemmas: *the*, *cat*, *be*, *teach*, *two*, *lesson*. The point of using the lemmas, rather than the words as used in the text, is to avoid having many different versions of the same phrase. In this example, we are much more likely to find significant statistics about the lemmatized *teach lesson*, than the original *taught lessons*. The downside is that it is not always possible to change inflections in a collocation, and expect the result to also be a collocation of the same meaning. For instance, we *hold hands with* someone rather than *hold hand with* them, but the lemma form of both versions is *hold hand with* and so they are indistinguishable by a tool using lemmatized text.

2.6 Collocation assistants

A number of tools to assist the user with finding suitable collocations have been developed. While there is considerable variation, all these tools essentially work by first producing a set of variations of the phrase under investigation, then comparing these to a text corpus, selecting the most "typical" alternatives.

Park et al. (2008) use an elegant Bayesian formulation to describe the task of a collocation assistant in general terms, and it is worth repeating here.

Given a phrase e from the user, we produce a set of variations c_i. We define a function $P(c)$ which is the probability of finding the candidate phrase c in the language, this could either be estimated directly from an n-gram frequency table, or computed indirectly, for instance using a Markov model of the language. Next, we define a conditional probability function $P(e|c)$ which is the probability that the writer has used c instead of e. This is very difficult to estimate, but heuristic functions including such data as the degree of synonymity between words in e and c, or the edit distance between the two phrases, can be devised. Now, for a candidate phrase c we have:

$$P(c|e) \propto P(e|c)P(c)$$

$P(c|e)$ is the probability that the user actually meant c, when writing e. If this is significantly higher than $P(e)$ for some c_i, then the tool should suggest that the user change phrase e to c_i.

Shei Shei and Pain (2000) describe a tool for helping users with collocations,[7] finding alternative phrases by substituting words with synonyms from a dictionary. The authors also suggest that during classroom use of their tool, several pieces of information should be collected and saved: a list of accepted collocations, a list of common unacceptable collocations, and a dictionary of definitions. The purpose of the latter is to attempt to replace clumsy expressions (such as *leave the ground*) with common collocations (*take off*).

AwkChecker Park et al. (2008) present *AwkChecker*, an "assistive tool for detecting and correcting collocation errors." This tool checks in real time if the phrase that the user is typing has a more common variant, using n-gram statistics. Unlike most other tools, it works with entire phrases of up to 5 words, and considers errors such as word order inversion, omitted prepositions as well as the kind of miscollocations detected by most other collocation assistants, where a word has been replaced with a semantically close word (which are found through WordNet (Fellbaum, 1998)).

Liu Lui et al. (2008) constructed a tool to suggest improvements to verb-noun collocations in English, using WordNet (Fellbaum, 1998) for semantic similarity, Mutual Information to measure collocation strength, and (their main innovation) using *collocation clusters* (see section 3.4.3) in evaluating candidate expressions. While this approach is interesting, Lui et al. (2008) only performed limited tests, insufficient to establish if the use of collocation clusters did in fact have an effect on the precision of their tool.

[7] It is unclear if it was ever successfully implemented and tested. A literature search turns up nothing, and attempts to contact the authors failed.

Futagi Futagi et al. (2008) use part of speech tagging and finite-state parsing to extract and analyze collocation candidates of four different types from an English text. In a manner similar to that of Park et al. (2008), a set of variations (with different articles, inflections and synonyms) is generated from each such phrase, and the frequency of candidates is looked up in a billion-word corpus. While words are not lemmatized (see section 2.5.4), generating variations with different inflections provides a similar function. One of the main features, absent in most other collocation assistants, is the use of spell checking and simple regular expressions to find collocation candidates. This results in a more robust tool, since a second-language learner likely to produce many miscollocations may also be assumed to make grammatical or spelling errors that could throw more fragile methods off track.

Writing Assistant Chang et al. (2008) created *Writing Assistant*, a tool that analyzes verb-noun collocations in English text, using a Chinese-English bilingual dictionary to check if replacing a word with another word sharing the same Chinese translation results in a stronger collocation. The authors show that around 84% of verb-noun miscollocations in a corpus of Taiwanese learners of English, can be improved by looking at words with the same Chinese translation. Their focus on verb-noun collocations is justified by citing studies showing that 87% of miscollocations in this corpus were of the type verb-noun, and in 96% of those, a better collocation could be obtained by changing the verb.

Stage	**Example**
Original text	*He did serious crimes.*
Tagged, lemmatized	*he* (pronoun) *do* (verb) *serious* (adjective) *crime* (noun)
Word pairs	*do crime* (and *serious crime*)
Similar candidates	*do crime*, *make crime*, *perform crime*, *commit crime*
Association strength	*do crime* (-1.40), *make crime* (-0.34), *perform crime* (7.24), *commit crime* (105.1)
New text	*He committed serious crimes.*

Table 8: The different stages of Antiskum, demonstrated on an English sentence. Numerical values are fictional, but realistic.

3 Method

I have designed and implemented a collocation assistant for Swedish text, called *Antiskum*. Its main characteristics are as follows:

- Using statistics from a 75 million word corpus.
- Corpus and user input lemmatized and tagged with parts of speech using Granska Tagger (Carlberger and Kann, 1999).
- Corpus and user input parsed with MaltParser (Nivre et al., 2007), although a custom finite-state parser is available as well.
- Able to process verb-noun (direct object) and adjective-noun collocations.
- Using *Folkets synonymlexikon* (Kann and Rosell, 2005) and a Random Indexing (Sahlgren, 2005) model to find semantically similar alternative phrases.
- Using Mutual Information and log-likelihood association measures to judge collocation strength.

In the following sections, I will attempt to justify the choices outlined above, and provide some further detail.

3.1 Overview

The function of Antiskum is summarized in table 8. First the text is parsed, lemmatized and part-of-speech tagged by Granska Tagger (second row). Then MaltParser or the finite-state parser is used to extract verb-noun or adjective-noun word pairs (third row). With the help of a synonym dictionary, Folkets synonymlexikon, we find a number of similar word pairs (fourth row) and compute association measures for the original and variant word pairs (fifth row). Finally, the best pair is fitted back into the sentence (sixth row).

A more detailed description of this process, along with a number of possible additions, will be given below.

3.2 Corpus

The corpus is compiled from three different text collections:

- 18,854,837 words: PAROLE[8], mostly newspaper articles from 1976–1997 and novels from 1976–1981.
- 15,514,145 words: A collection of WWW news articles, collected during early 2009.
- 40,991,680 words: All articles from the Swedish Wikipedia[9].

A *word* here refers to one token identified by Granska Tagger (Carlberger and Kann, 1999), excluding punctuation (defined as tokens tagged as **mad**, **mid** or **pad**). In total there are 75,360,662 words.

As we discussed in section 2.5.1, quantity is important when gathering collocation statistics, and it was my primary criterion in selecting the sources. PAROLE, the news corpus and Wikipedia were the only large bodies of Swedish text that were freely available and not obviously unsuitable. The Runeberg[10] and Gutenberg[11] projects mainly host older works (due to copyright restrictions). Since spelling, grammar and collocation usage change over time (Malmgren, 2002) I chose not to include these sources.

3.3 Text processing

We need to process written Swedish text at two points: first when gathering statistics for the word and word pair frequency database and for our Random Indexing model, and again when processing the end user's text.

3.3.1 Corpus processing

The very first step of processing is to produce plain text files. In the case of the PAROLE corpus, which has been annotated with an incompatible set of part of speech tags, this simply meant removing the POS tags. Wikipedia can be downloaded as an XML file containing information about articles, as well as the text of the articles in the MediaWiki markup format used by Wikipedia. A set of regular expressions was constructed in order to extract raw text from this. Due to syntax errors, direct quotes from old or foreign texts and vandalism by Wikipedia users, a small portion of undesirable material remains. This is not expected to cause any significant problems.

Next, Granska Tagger (Carlberger and Kann, 1999) was used to split the raw text files up into words, lemmatize and add part of speech tags to them. The implications of this decision were discussed in section 2.5.4.

The output of Granska Tagger, which uses a modified version of the SUC tagset, was then converted to the CoNLL data format[12] with original SUC tags.

[8] http://spraakbanken.gu.se/parole/

[9] http://sv.wikipedia.org/

[10] http://runeberg.org/

[11] http://www.gutenberg.org/browse/languages/sv

[12] http://nextens.uvt.nl/depparse-wiki/DataFormat

Finally, MaltParser (Nivre et al., 2007) version 1.2 was used to perform a full dependency parsing of the corpus. MaltParser had been trained using a SUC-tagged version of Talbanken[13] obtained from Joakim Nivre, since the default Talbanken tagset is incompatible with the SUC-based tagset used by Granska Tagger.

The parsing step was by far the most time-consuming, with each million words requiring about 4 hours of CPU time, the entire corpus required nearly two weeks to process.

What follows is an example of a single sentence from the corpus ("She started dancing ballet at the age of four").

```
1   Hon      hon     PN   PN   UTR|SIN|DEF|SUB 2   SS
2   började  börja   VB   VB   PRT|AKT         0   ROOT
3   dansa    dansa   VB   VB   INF|AKT         2   OO
4   balett   balett  NN   NN   UTR|SIN|IND|NOM 3   OO
5   vid      vid     PP   PP   _               2   TA
6   fyra     fyra    RG   RG   NOM             7   DT
7   års      år      NN   NN   NEU|PLU|IND|GEN 8   DT
8   ålder    ålder   NN   NN   UTR|SIN|IND|NOM 5   PA
9   .        .       MAD  MAD  _               2   IP
```

The different columns are:

1	Index within sentence
2	Token, directly from input text
3	Lemma
4–5	Part of speech tag (redundant)
6	Detailed POS information
7	Head
8	Syntactic relationship to head

In row 4 of our example, we find the noun (**NN**) *balett* (ballet). Its head is at row 3, containing the verb (**VB**) *dansa* (dance), and the **OO** syntactic relationship indicates that *balett* is the direct object of *dansa*. This forms a verb-noun pair, one of the collocation types we are interested in.

3.3.2 Gathering statistics

An SQLite[14] database is used to store statistics about different words.

For each word, its lemma, POS tag and number of occurrences in the corpus are stored.

For each verb-noun or adjective-noun pair, defined as a word and its head-word (according to MaltParser), the lemmas and POS tags of the words, their syntactic relation, and number of occurrences in the corpus are stored.

3.3.3 Collocation types

Antiskum only considers verb-noun (where the noun is a direct object) and adjective-noun pairs. A previous version also used verb-adverb pairs. Why these particular types?

[13] http://w3.msi.vxu.se/~nivre/research/Talbanken05.html

[14] http://www.sqlite.org/

In section 2.1.4 we discussed different types of collocations, and saw that several tools deal exclusively with verb-noun collocations, something that Nesselhauf (2005) also limits herself to in her in-depth study of miscollocations by advanced learners of English. Futagi et al. (2008) cite one study, showing that 87% of collocation errors in a learner corpus were verb-noun collocations, with the verb being the problem in 96% of those cases. That the verb is the most common problem is confirmed by Nesselhauf (2005), who found a verb substitution to be the problem in 52% of verb-noun miscollocations in a different corpus.

While the case for including verb-noun collocations is strong, adjective-noun collocations have been investigated less. They are fairly common, usually easy to extract from a text, so I follow Futagi et al. (2008) in dealing with adjective-noun collocations as well.

There are two main reasons why the initial support for verb-adverb pairs was dropped. First, these pairs are much less frequent than either verb-noun or adjective-noun pairs. Second, Folkets synonymlexikon does not separate different parts of speech, which means that there is some confusion between adverbs and neuter-inflected adjectives, which interferes with the function of Antiskum.

3.3.4 Finite-state parsing

MaltParser (Nivre et al., 2007) is a resource-intensive program, so is desirable to find an alternative solution to the problem of finding the verb-noun and adjective noun word pairs that we are interested in.

Futagi et al. (2008) use finite-state parsing, implemented as a set of regular expressions defined over the part-of-speech tags of a text to extract the word pairs of interest to their collocation assistant, so I have opted for a similar solution.

The following patterns are used for verb-noun and adjective-noun pairs, with the first and the last word in each pattern being paired up:

```
verb (determiner|adverb|adjective)* noun
adjective (adjective|conjunction)* noun
```

3.4 Finding semantic relations

One of the essential tasks when looking for alternatives to a miscollocation is determining the semantic similarity between the original phrase and possible alternatives.

3.4.1 Synonym dictionaries

Folkets synonymlexikon (Kann and Rosell, 2005) appears to be the only decent Swedish dictionary of synonyms freely available in electronic format, so its inclusion in Antiskum was an obvious choice.

Many of the English collocation assistants in section 2.6 use the English WordNet (Fellbaum, 1998). There is a corresponding, but much smaller Swedish WordNet (Viberg, 2002), but this is not freely available and since Folkets synonymlexikon is rather complete, I did not consider it necessary to include the Swedish WordNet.

I also considered SALDO (Borin and Forsberg, 2009) in the final stages of the project, but its lack of part-of-speech distinction, unusual semantic model, as well as a lack of time on my part prevented further investigation.

3.4.2 Random Indexing

One of the anonymous reviewers of a previous article about the Antiskum project (Östling and Knutsson, 2009) suggested using some vector space model to complement Folkets synonymlexikon, and I took this advice to heart, using Random Indexing for my experiments. The theory behind Random Indexing was discussed in section 2.4.2. Here we will detail the particular Random Indexing model used in Antiskum.

The vectors used are of length 1,500 and each index vector has 4 elements that are 1, and 4 elements that are -1.

A *word* is represented by a lemma and a part of speech tag, so that e.g. (*insult*, noun) and (*insult*, verb) are distinct words. A *context* consists of the following:

lemma Lemma of context word
POS Part of speech of context word
before? 1 if the context word comes *before* the word in the sentence,
0 if it comes after
head? 1 if the context word is the head of the word,
0 if the word is the head of the context word

We can illustrate this with the phrase *eat tasty fruit*. Here, the head of *tasty* is *fruit*, and the head of *fruit* is *eat*. The two contexts for *fruit* will therefore be: (*tasty*, adjective, 1, 0) and (*eat*, verb, 1, 1).

3.4.3 Collocation clusters

Fellbaum (1998) use *collocation clusters* when ranking suggested changes to a verb-noun collocation. The idea is to figure out what other words are in the range of the same lexical functions as the collocate. Given a collocation candidate (b, c), with basis b and collocate c, we search in our word count database for a set B, containing other bases so that for $w \in B$, (w, c) is common. Next, we search for a similar set C, containing other collocates so that for $u \in C$, (b, u) is common.

There is usually some subset $C' \subset C$ and $B' \subset B$ such that for $w \in B'$ and $u \in C'$, (w, u) tends to be common. This can indicate that one or more lexical functions exist whose domains include B', and ranges include C'.

As an example, consider the adjective-noun pair *powerful tea*. We might then have $B = \{$ *engine*, *idea*, *factor*, *man*, *tool* $\}$, that is, nouns that collocate with *powerful*. And $C = \{$ *black*, *strong*, *hot* $\}$, adjectives that collocate with *tea*.

We can immediately see that *strong* $\in C'$, since *strong tea*, *strong factor* and *strong man* are all quite common. The explanation for this is that *strong* and *powerful* in this case are nearly synonymous and part of the range of the lexical function describing someone or something influential.

However, other (irrelevant) lexical functions are also involved. For instance, we have *hot engine*. Both tea and engines can be *hot*, but they have little to do with *powerful* and *strong*. From this we can see that while collocation clusters have the potential to be useful in narrowing down the set of likely candidates when we are trying to find a better collocation, they may also introduce misleading information.

Antiskum can use collocation clusters, the metric used for each $u \in C$ is the proportion of $w \in B$ where (w, u) is common. A value of 1 indicates that w is common with *all* of the words in B. This is an uncommon situation, in experiments, this factor typically is below 0.2 and quite often 0 for acceptable choices of u.

3.5 Using Antiskum

As the previous sections show, a great deal of work has to be done before we even get to the part of Antiskum that the end user will see. Now we turn to this final component, whose task is to find and evaluate verb-noun and adjective-noun pairs in a Swedish text, based on the different measures described above.

The main question is: given all this information about the different candidate word pairs, how do we best choose one? My first approach was to use decision trees (Mitchell, 1997). It turns out that with tens of unacceptable suggestions for each acceptable one, always guessing that a suggestion is unacceptable is a very accurate prediction. Only in two cases, when Folkets synonymlexikon indicates that the collocates are synonymous or when both the log-likelihood and Mutual Information association measures are very high, "acceptable" becomes a better guess.

The next attempt was to assume that there is a function $f(M, x)$ for the array of metrics x and a model M, such that $f(M, x)$ is higher when x represents an acceptable suggestion than otherwise.

$$f(M, x) = M_1 x_{fs \neq 0} + M_2 x_{ri} + M_3 x_{mill} + M_4 x_{mi} + M_5 x_{cc \neq 0} + M_6 x_{cc}$$

Where $x_{mill} = \text{sgn}(x_{mi}) \sqrt{\|x_{ll} x_{mi}\|}$, and $x_{fs \neq 0}$ is 1 if Folkets synonymlexikon lists the collocates as synonyms regardless of the synonymity level, 0 otherwise. $x_{cc \neq 0}$ (collocation cluster measure) is defined analogously. The other variables represent the numeric values of their respective metrics.

The square root of the Mutual Information and log-likelihood product was chosen on the basis of experiments indicating that this gives a better result with the linear function f above, than using the product directly.

The value of M was determined by a randomized local search algorithm, with the goal to maximize the number of cases in the test set, where an acceptable suggestion x was in the n-best $f(M, x)$.

We have described how to find alternatives to a known miscollocation, but how do we know which word pairs are miscollocations? The approach used in Antiskum is to simply treat each verb-noun or adjective-noun pair as a suspected miscollocation, and look for better alternatives. If the original pair is the most likely candidate, then there is no need to bother the user. If there are better candidates, these are mentioned.

4 Results

There are several possible usage scenarios for a collocation assistant, each placing different demands on our tool.

Improving the prose of a language learner. This appears to be the most common use for collocation assistants, which have mostly been constructed with ESL (English as a second language) users in mind. In this case, we can expect a significant minority of collocation candidates to be miscollocations.

Correcting a known miscollocation. If a learner wants to find a more acceptable version of a known miscollocation, perhaps a construction that exists in the speaker's native language but not in Swedish (such as *make bed* vs. **göra säng*), our task is simplified by the fact that we can exclude the original word pair from consideration, and focus on the other candidates.

Checking a well-written text. When checking a text by a competent native writer, miscollocations are rare. Our main task as designers of a collocation assistant is to reduce the number of false alarms. However, since a collocation often has other acceptable variants, this does not necessarily mean that the tool ought to be entirely silent.

4.1 Miscollocation list

I have constructed a list of 200 verb-noun miscollocations along with their acceptable variants. Ideally, these examples should be taken from a corpus, as is done by e.g. Lui et al. (2008). Finding a sufficient number of suitable miscollocations is, however, a very time-consuming tasks, and I do not have access to any such data for Swedish. The vast majority of the miscollocations have been chosen as mistranslations from English collocations, in an attempt to make them somewhat less arbitrary and prone to being biased towards "easy" cases than if they were to be chosen entirely arbitrarily by the author. The complete list is available in appendix B.

The main purpose of this test is to determine how to weigh all the information we have about the different collocation candidates, such as synonymity, Mutual Information and log-likelihood scores, according to $f(M, x)$ in section 3.5.

Table 9 shows the results. The features are those listed in section 3.5, where *fs* and *cc* correspond to the boolean variables $x_{fs\neq0}$ and $x_{cc\neq0}$, and the others are real-valued.

Keeping in mind that the sample size is 200, we can see at a glance that the last three rows contain no important differences. From this we can conclude that including collocation cluster information (*cc*) does not significantly improve the results, and apart from a slight improvement in the *Top 3* experiments, neither does using the Random Indexing score (*ri*),

We can make two observations that may be of some significance: the Mutual Information and log-likelihood product seems to be better than just the Mutual Information, and the use of Folkets synonymlexikon seems to improve the results as well. However, in the last two columns (which include the original miscollocation in the candidate list), the difference is much smaller.

Table 9: Percentage of the 200 miscollocations in our test set where Antiskum gave an acceptable collocation as the best (*Best*) alternative or among the best three (*Top 3*) alternatives, when sorted by $f(M, x)$ for the best model M found. In the *Best** and *Top 3** experiments, the original miscollocation was removed from consideration, making the task somewhat easier.

Features	*Best**	*Top 3**	*Best*	*Top 3*
mi	38.5	64.5	38.5	62.0
mill	47.5	72.5	47.5	70.0
fs, mi	50.5	70.5	42.5	66.0
ri, mill	47.5	75.0	47.5	72.5
fs, mill	55.0	77.0	48.5	72.0
fs, ri, mill	55.0	79.0	48.5	75.0
fs, ri, mill, cc	56.0	79.0	48.5	75.0

We can also see that if we accept that an acceptable phrase simply is among the top three candidates when sorted by $f(M, x)$, the result is 20 to 25 percentage points better compared to only accepting cases where the top candidate is acceptable.

A precise analysis of these numbers is difficult. First of all, the verbs and nouns in the miscollocations are often used more than once, which means the different test cases are not completely independent of each other. Second, some miscollocations are inherently easy to correct, for instance when an acceptable verb is the strongest collocate of the noun, or when an acceptable verb is listed in the synonym dictionary.

As we can see in the second row of table 9 (*mill*), as many as 47.5% of the miscollocations (95 cases) could be corrected using collocation strength alone. The figure for collocation strength and synonymity (*mill, fs*) is 55.0% (110 cases) which means that only 15 of the 105 remaining cases have been solved.

In other words, the performance depends heavily on the types of miscollocations in the sample, and these results will likely differ considerably from the performance in any real-life situation.

4.2 Professional prose

A 5 164 word corpus of encyclopedia articles and opinion pieces was processed with Antiskum, once using MaltParser and once using the finite-state parser described in section 3.3.4.

This experiment is designed to test two important aspects of Antiskum: its ability to suggest reasonable synonyms to already acceptable phrases, and the rate of false alarms.

Using Folkets synonymlexikon and the *mill* association measure (see section 3.5) gives a very large number of suggested changes (300 for MaltParser, 379 for the finite-state parser), most of which have a synonymity level of zero. In other words, the strongest collocate of the noun is chosen, whether or not it is relevant—and it usually is not. This results in an unmanageable amount of bad suggestions.

Table 10: Results of Antiskum on 5 164 words of professional prose, using either MaltParser or the finite-state parser.

	MaltParser	*Finite-state*
Nr. of pairs	480	569
Suggestions	25	30
Acceptable	18	21
Unacceptable	7	9

If we require non-zero values from Folkets synonymlexikon, things look very different (table 10). This makes the algorithm equivalent to version summarized in section 3.1.

Suggestions for changes are deemed acceptable if the new phrase is a collocation of the same meaning as the original phrase. Different degrees of the same basic meaning (such as replacing *big* with *enormous*) are not considered acceptable.

The main reason why the finite-state parser finds more word pairs than MaltParser seems to be that MaltParser omits some actual pairs. In none of the cases investigated in table 10, either parser formed a verb-object or adjective-noun pair out of unrelated or otherwise related words. When checking larger samples, however, one occasionally finds such erroneously formed word pairs.

4.3 Learner essays

Two short essays (587 words in total) from a Swedish learner (native speaker of Mandarin and French) were analyzed. No clear miscollocations are present in these essays. This sample is unfortunately much too small to draw any firm conclusions, but the results are presented and briefly discussed below.

The essays contain errors of spelling and grammar, which pose additional challenges. In fact, several erroneous word pairs were found in these brief texts. For instance, a spelling error (*natur resursen* for *naturresurser*) caused the finite-state parser to extract the verb-noun pair *spara natur* (*save nature*), which Antiskum then suggested should be changed to *bevara natur* (*preserve nature*). A good answer, unfortunately to the wrong question.

MaltParser finds 32 word pairs, and makes 4 suggestions. Two are acceptable, two are not.

The finite-state parser finds 46 word pairs and makes 6 suggestions. Four are acceptable, two are not.

5 Conclusions and recommendations

What can we learn from the successes and failures of this project? What directions should future research into collocation assistants take?

5.1 Corpus

Since most individual collocations occur very sparsely, a large corpus is essential. This can be difficult to obtain, especially for a language like Swedish with only some ten million speakers. Large quantities of text are of course technically available, but for copyright reasons and a vague hope of some future profit, these are not distributed.

Collecting text from the Internet is convenient and can yield considerable amounts of data. Unfortunately, it is not trivial to extract plain text from this data, and it is even more difficult to tell if a text constitutes a good sample of the language. Much of the text publicly available on the World Wide Web would not make a very good model for insecure writers.

During my experiments, I never noticed any mistakes by Antiskum that could be attributed to deficiencies in the corpus, so corpus quality does not appear to be a great concern for this design. In case a concordancer is included, as in the early version of Antiskum (Östling and Knutsson, 2009), one might want to screen the text presented to the user. The growing body of research on automated essay grading could perhaps be applied for this purpose.

5.2 Statistics

As we have seen, the popular Mutual Information association measure has the disadvantage of not taking uncertainty into account. Better results can be obtained by using the product of the Mutual Information and log-likelihood measures, as described in section 3.5.

I implemented and evaluated the *collocation clusters* of Lui et al. (2008), as described in section 3.4.3. While their initial experiment (using a very small sample) showed some promise, no significant improvement was found in Antiskum when enabling collocation clusters.

5.3 Parsing

One of the unique features of Antiskum is its use of a full parser, MaltParser. Other collocation assistants use simple methods (see section 2.5.3): either no parsing at all beyond the word level, or a simple finite-state parser to locate a small set of sentence patterns.

What does this buy us? Is it worth it, considering the significant cost in processing time by the parser?

As the experiments in sections 4.2 and 4.3 show, using MaltParser for processing a text actually results in a somewhat lower word pair detection frequency compared to the finite-state parser, without any significant improvement in accuracy. This is in spite of the very simple regular expressions used (see section 3.3.4). More sophisticated rules could probably improve the record even further.

So while using MaltParser for processing user texts seems to be a bad idea, how about the other uses of MaltParser in this project? The entire corpus has been parsed, and this forms the basis of both the word pair frequency statistics and the Random Indexing model used (see section 3.4.2). Since Random Indexing turned out not to be of much use, and the word pair statistics are not greatly affected by a small difference in detection frequency, we are forced to conclude that using a full parser does not seem to benefit a collocation assistant of the current design.

I would recommend other designers of collocation assistants not to use a complex, general parser. A set of relatively simple regular expressions can do a good job, at least for the verb-noun and adjective-noun pairs used for this project, and apparently also for the other patterns used by Futagi et al. (2008).

5.4 Semantic considerations

Let us step back for a moment to look at our task from a human perspective. Suppose that we find a phrase, *he did several crimes* for instance, which sounds a bit awkward. *Did* sounds out of place, and we would not have to think very long to figure out that the author probably meant to use the similar but more common *commit*.

So far, so good. This is essentially how Antiskum and other collocation assistants work, by finding similar (according to a synonym dictionary) phrases that sound more familiar (according to statistics gathered from a text corpus).

There are more difficult cases. “Their house lost fire”—what could this mean? Did the fire in their house die out? If the writer’s native language is Chinese, and we know that 失火 *shī huǒ* (literally *lose fire*) means to catch fire, it is not difficult to guess that this is a case of interference from the writer’s native language. Chang et al. (2008) focus on this phenomenon to create a tool aimed specifically at Chinese learners of English. Given a good enough dictionary, this type of miscollocation can also be corrected automatically.

Things can get even more difficult, unfortunately. Consider the phrase *ta självmord* (*take suicide*), a mix-up of *ta sitt liv* (*take one’s life*) and *begå självmord* (*commit suicide*) which is becoming quite common in informal Swedish. The verb gives little hint as to the intended meaning of the phrase. Looking at common collocates for *självmord* (*suicide*), we might find *överväga* (*contemplate*) and *begå* (*commit*), but which one to choose? As a human, we would look at the context. Did he do something afterwards? In that case he probably didn’t *commit* suicide. Although... Maybe it was just a dream?

These are considerations mostly beyond the reach of today’s computers, and this is the fundamental reason why we can not expect to always be able to improve the collocations in a text automatically.

Leaving automatic improvement aside for a moment, how about just *detecting* miscollocations? If we can do this, then our tool can simply show more common variations of that phrase and let the user decide which is closest to the intended meaning, perhaps with the help of a concordance.

If we look at collocations as constructions (see section 2.1.2) that are customarily used to express a certain concept or idea, then a miscollocation is an attempt to express that idea without using the usual construction, thereby producing a phrase that might seem clumsy and more difficult than necessary to understand. But the proof of the miscollocation *is* the corresponding collocation. This is why we can say that *powerful tea* is a miscollocation (of *strong tea*) while a non-collocation like *flammable tea* is certainly unusual, but lacks a more common construction to express the same meaning.

In the end, if we do not have a better suggestion, it is better to assume that the writer really meant what he or she wrote.

5.5 Error modeling

Let us go back to the Bayesian model recounted in section 2.6. $P(e|c)$ should represent the probability that the user writes e when a competent writer would use c. Antiskum simply uses Folkets synonymlexikon to estimate this function, but as we have just seen from expressions like *lose fire* and *take suicide*, this problem is much more complex. After all, $P(e|c)$ depends on the native language, amount of language exposure and other factors in the mind of the user.

5.5.1 Brute force

When faced with such a complex function, we might feel the urge to simply measure its values, as Shei and Pain (2000) indeed suggest. Would this be practical?

Nesselhauf (2005) found a rate of 4.85 verb-noun miscollocations per 1000 words in a corpus of advanced English learners. If we assume that a language student writes ten short essays of 200 words each every year, this amounts to an annual production of 2,000 words per year and student, which gives about 10 verb-noun miscollocations per student and year if Nesselhauf's findings can be generalized.

How many pairs of (e, c) would we need to provide a useful estimate of $P(e|c)$? A few tens or hundreds of thousands of pairs would probably cover most common miscollocations. With ten pairs per student and year, a few thousand students would be sufficient to gather this amount of data in a reasonable time.

While such an undertaking is obviously outside the scope of my project, collocation assistants could benefit along with other fields within computational linguistics and computer-assisted language learning from a systematic effort to collect annotated learner essays.

5.5.2 Lexical functions and semantic relations

There are various ways to model the causes of miscollocations. One common assumption, which Antiskum is based upon, is that $P(e|c)$ is roughly proportional to the degree of synonymity (defined in various ways) between the words in e and c. To be optimally useful, a synonym dictionary should:

- Be comprehensive.
- Separate parts of speech, in order to decrease the number of irrelevant suggestions, if the collocation assistant provides part of speech information.
- Separate word senses, in order to decrease the number of irrelevant suggestions, if the collocation assistant performs word sense disambiguation.

Instead of a synonym dictionary, some projects use measures such as WordNet distance, or whether the words share a translation in the user's native language. In these cases, we would desire about the same properties as from a synonym dictionary.

With a dictionary of lexical functions, we can increase $P(e|c)$ if a corresponding pair of words in e and c belong to the range of the same lexical function. For instance, *powerful* and *strong* both belong to the function describing a high intensity, so $P(\textit{powerful tea}|\textit{strong tea})$ should be fairly high.

5.6 Phrases

While Antiskum assumes that a verb-noun miscollocation can always be corrected into a verb-noun collocation, this is not the case in reality. Park et al. (2008) consider a few structural changes to phrases, such as word reordering, and the insertion or deletion of prepositions. But when it comes to making mistakes, there is no end to human creativity.

Consider for instance the word *self-kill*, a direct translation from Chinese which seems to be an intransitive verb formed by adding the *self-* prefix to *kill*. We would put this in proper English as *commit suicide*, a transitive verb followed by a direct object. One could easily guess that the meaning is the same, but the structure is completely different from *self-kill*.

As humans, when we try to correct someone's clumsy or faulty language, we first try to understand a phrase through its contents and its context, then we formulate our own phrase to express the same meaning. For this task we use our mental library of constructions, and could apply collocations, idioms, proverbs or famous quotations as we see fit. In a way, our task is one of translation: from bad language, into good language.

This, I think, is the ultimate goal of collocation assistants.

References

Borin, L. and Forsberg, M. (2009). All in the family: A comparison of saldo and wordnet. In *Proceedings of the Nodalida 2009 Workshop on WordNets and other Lexical Semantic Resources - between Lexical Semantics, Lexicography, Terminology and Formal Ontologies.*

Carlberger, J. and Kann, V. (1999). Implementing an efficient part-of-speech tagger. *Software – Practice and Experience*, 29:815–832.

Chang, Y.-C., Chang, J. S., Chen, H.-J., and Liou, H.-C. (2008). An automatic collocation writing assistant for Taiwanese EFL learners: A case of corpus-based NLP technology. *Computer Assisted Language Learning*, 21:283–299.

Cinková, S., Podveský, P., Pecina, P., and Schlesinger, P. (2006). Semi-automatic building of swedish collocation lexicon. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1890–1893.

Domeij, R., Knutsson, O., Carlberger, J., and Kann, V. (1999). Granska - an efficient hybrid system for Swedish grammar checking. In *NoDaLiDa, December 1999*, pages 49–56.

Evert, S. (2005). *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. PhD thesis, Universität Stuttgart.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. MIT Press.
`http://wordnet.princeton.edu/`.

Futagi, Y., Deane, P., Chodorow, M., and Tetreault, J. (2008). A computational approach to detecting collocation errors in the writing of non-native speakers of English. *Computer Assisted Language Learning*, 21:353–367.

Goldberg, A. E. (2003). Constructions: a new theoretical approach to language. *TRENDS in Cognitive Sciences*, 7(5):219–224.

Guo, S. and Zhang, G. (2007). Building a customised Google-based collocation collector to enhance language learning. *British Journal of Educational Technology*, 38:747–750.

Johansson, S. (2001). *Kollikon – Frasidentifikation och -extrahering*. Master's thesis, Göteborgs universitet.
`http://folk.uio.no/danielr/Kollikon.pdf`.

Kann, V., Domeij, R., Hollman, J., and Tillenius, M. (1998). Implementation aspects and applications of a spelling correction algorithm. Technical report, Numerical Analysis and Computing Science, Royal Institute of Technology. S100 44, Stockholm, Sweden.

Kann, V. and Rosell, M. (2005). Free construction of a Swedish dictionary of synonyms. In *NoDaLiDa 2005, Joensuu.*

Krenn, B. and Evert, S. (2001). Can we do better than frequency? A case study on extracting PP-verb collocations. In *Proceedings of the ACL Workshop on Collocations*.

Lewis, M., editor (2000). *Teaching Collocation: Further Developments in the Lexical Approach*. Language Teaching Publications.

Lin, D. (1998). Extracting collocations from text corpora. In *First Workshop on Computational Terminology*, pages 57–63.

Lui, A. L.-E., Wible, D., and Tsao, N.-L. (2008). Automated Suggestions for Miscollocations. In *Proceedings of the 46th Annual Meeting of the ACL*, pages 47–50.

Malmgren, S.-G. (2002). *Begå eller ta självmord? – Om svenska kollokationer och deras förändringsbenägenhet 1800–2000*. University of Gothenburg. Report 15 in the ORDAT series: "Det svenska ordförrådets utveckling från artonhundra till tjugohundra."
`http://spraakdata.gu.se/ordat/pdf/ORDAT15.pdf`.

Manning, C. and Schütze, H., editors (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.
`http://nlp.stanford.edu/fsnlp/`.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Nesselhauf, N. (2005). *Collocations in a Learner Corpus*, volume 14 of *Studies in Corpus Linguistics*. John Benjamins Publishing Company.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., and Marsi, E. (2007). MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.
`http://maltparser.org/`.

Östling, R. and Knutsson, O. (2009). A corpus-based tool for helping writers with Swedish collocations. In *Proceedings of the Nodalida 2009 Workshop on extracting and using constructions in NLP*. SICS Technical Report T2009:10
`http://www.sics.se/~mange/papers/constructions_workshop.pdf`.

Park, T., Lank, E., Poupart, P., and Terry, M. (2008). Is the sky pure today? AwkChecker: an assistive tool for detecting and correcting collocation errors. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 121–130, New York, NY, USA. ACM.

Pearce, D. (2002). A comparative evaluation of collocation extraction techniques. In *Third International Conference on Language Resources and Evaluation*.

Petrović, S., Šnajder, J., Bašić, B. D., and Mladen, K. (2006). Comparison of collocation extraction measures for document indexing. *Journal of Computing and Information Technology*, 14(4):321–327.

Quasthoff, U. and Wolff, C. (2002). The Poisson Collocation Measure and its Applications. In *Workshop on Computational Approaches to Collocations*.

Sahlgren, M. (2005). An Introduction to Random Indexing. In *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering.*

Shei, C. C.-C. (2005). Plagiarism, Chinese Learners and Western Convention. *Taiwan Journal of TESOL*, 2(1):97–113.

Shei, C. C.-C. and Pain, H. (2000). An ESL Writer's Collocational Aid. *Computer Assisted Language Learning*, 13(2):167–182.

Smadja, F. (1993). Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.

Viberg, Å. (2002). The Swedish WordNet Project. In *Proceedings of Euralex 2002, Copenhagen University*, pages 407–412.

A Glossary

Basis The "fixed" word of a collocation, carrying most of the collocation's meaning. Assumed by Antiskum to be the noun in the verb-noun and adjective-noun pairs considered. The other word is the **collocate**. See section 2.1.4.

Collocate The "variable" word of a collocation, as opposed to the **basis**. See section 2.1.4.

Collocation See section 2.1.

Finite-state parsing The process of extracting certain word patterns (such as the verb-noun and adjective-noun pairs checked by Antiskum) using regular expressions (equivalent to finite state automata) over the words and their part-of-speech tags.

Idiom Set phrase, whose meaning is not obvious from the constituent words. The line between idioms and **collocations** is blurry.

Lemma The uninflected form of a word. For instance, the lemma of *be*, *is*, *was*, *are*, etc. is *be*.

Parsing Identifying syntactical relationships between words in a sentence, e.g. finding the verb-object pair *eating-tomato* in the phrase "I am eating a small tomato." See section 2.5.3.

Part of speech (POS) Word class, e.g. noun, verb, adjective, etc. Part-of-speech tagging is the process whereby each word of a text is annotated with its (probably) part of speech.

B List of verb-noun miscollocations

The following list of 200 verb-noun miscollocations and the corresponding proper Swedish collocations was constructed by myself, primarily by translating English collocations literally and idiomatically into Swedish. While material from e.g. a learner corpus would have been preferable, gathering and processing this would have required time and materials currently unavailable to me.

Note that some verbs occur in several different conjugations, this is due to the faulty lemmatizer of Granska Tagger.

Acceptable	**Not acceptable**	**Nouns**
snacka, prata	*tala*	**skit**
prata	*snacka, tala*	**strunt**
föra, driva	*leda*	**politik**
föra	*driva*	**krig**
göra, koka, brygga	*laga*	**te, kaffe**
rensa	*rena*	**fisk**
fatta	*gripa*	**penna**
klippa	*trimma*	**hår**
fatta	*tappa, förlora*	**eld**
känna	*kunna*	**folk**
förstöra	*ruinera*	**liv, rykte**
hålla	*göra*	**tal**
teckna, ingå, sluta, slöt, slutit	*göra*	**avtal**
väcka	*börja, starta*	**debatt**
väcka	*attrahera*	**uppseende**
väcka	*hetsa*	**ilska**
hålla	*bevara, behålla*	**djur, boskap**
täcka, omfatta	*omsluta*	**område**
lägga	*sätta, ställa*	**arm**
behärska	*övervinna*	**konst**
bädda	*göra*	**säng**
äta	*ha*	**frukost, lunch, kvällsmat, middag, mat, måltid**
bryta	*förstöra*	**ben, gren**
spänna	*dra*	**båge**
utgöra, bilda	*konstituera*	**gren**
knäppa	*fästa*	**knapp**
rikta	*sikta*	**kamera**
blanda	*flytta*	**kort**
dra	*rycka*	**vagn**
bilda	*utgöra, konstituera*	**kedja**
föda	*bära*	**barn, dotter, son**
uppfostra	*odla*	**barn**
vända	*rikta, vrida*	**rygg**
åka	*rida*	**bil, buss**
vrida	*vända, rikta*	**klocka**

Acceptable	Not acceptable	Nouns
ställa	*sätta, lägga*	**klocka**
inleda	*öppna, starta*	**samtal**
skjuta	*trycka*	**stol**
väcka	*lämna*	**åtal**
ha	*närvara*	**lektion**
hålla	*bevara*	**kyla, värme**
ha, driva	*springa, löpa*	**affär, företag**
ge, lämna	*skänka*	**samtycke**
bryta, avbryta	*klippa*	**förbindelse**
bryta, avbryta	*överge*	**tävling, match**
bryta	*överge*	**lopp**
utgöra, innebära, medföra	*posera, presentera, ställa*	**fara**
ägna, tillbringa	*spendera*	**dag, natt**
föra, hålla	*leda*	**debatt**
fatta, ta	*göra*	**beslut**
ställa	*göra, placera*	**krav**
rasta	*gå*	**hund**
förverkliga, realisera, fullfölja, uppfylla	*inse*	**dröm**
bära	*använda*	**klänning**
spela	*leka*	**dum**
spetsa	*skärpa, vässa*	**öra**
knäcka	*spräcka*	**ägg**
sätta	*ställ, lägga*	**stopp**
spara	*konservera*	**energi**
bilda	*starta*	**familj**
injaga, ingjuta	*inge*	**skräck**
ta	*debitera*	**avgift**
ha	*springa, löpa*	**feber**
sätta, lägga	*ställa*	**finger**
hissa	*lyfta, höja*	**flagga**
fylla	*tjäna*	**funktion**
fatta	*fånga*	**eld**
laga	*koka*	**mat**
spela	*leka*	**spel**
leka	*spela*	**lek**
hysa	*ha*	**agg**
klippa	*skära*	**gräs**
klippa	*skära*	**hår**
knäppa	*sluta*	**hand**
bärga	*skörda*	**skörd**
krossa	*förstöra*	**hjärta**
bryta	*förstöra*	**is**
göra, genomföra	*föra*	**utredning**
innebära, medföra, orsaka	*framkalla*	**ökning**
ha	*bära*	**ränta**
skratta	*ge*	**skratt**

Acceptable	Not acceptable	Nouns
stifta	*statuera*	**lag**
fälla, tappa	*släppa*	**löv**
dyrka	*pilla, peta*	**lås**
dra	*kasta*	**lott**
spela	*leka*	**match**
väcka, framkalla	*frammana*	**minne**
ana	*mistänka*	**oråd**
tjäna	*göra*	**pengar**
bestiga	*klättra*	**berg, topp**
spänna	*böja*	**muskel**
sprida, förmedla	*bära*	**nyhet**
avböja, avvisa, avslå	*vägra, neka*	**erbjudande**
uttrycka, framföra, uttala	*yttra*	**åsikt**
hålla, upprätthålla, bevara	*behålla*	**ordning**
tillfoga	*påtvinga*	**smärta**
sluta, slöt, slutit	*göra*	**fred**
ge, utfärda, bevilja	*medge, skänka*	**tillstånd**
vässa	*spetsa, slipa*	**penna**
sysselsätta, anställa	*använda*	**person**
sätta	*ställa, lägga*	**press**
driva	*föra, jaga*	**process, fråga, linje**
förvalta	*sköta*	**egendom**
väcka, orsaka, vålla, framkalla	*åstadkomma*	**protest**
avtjäna	*tjäna*	**straff**
hålla	*behålla*	**kvalitet**
ställa	*lägga, sätta*	**fråga**
inleda	*börja, starta*	**förhållande**
utöva, praktisera	*utföra*	**religion**
avslå	*neka, vägra*	**begäran**
åtnjuta	*avnjuta*	**respekt**
åtnjuta, ha	*avnjuta*	**anseende**
utlova, utfästa, utfäst, utfäste, utlysa, utfärda, erbjuda	*framföra*	**belöning**
koka	*laga*	**ris**
byta	*ändra*	**sida**
valla, vakta	*driva*	**får**
putsa	*polera*	**sko**
skotta	*skyffla*	**snö**
framföra	*uppföra*	**sång, låt**
hålla	*göra*	**anförande**
resa	*uppföra*	**sten**
berätta, dra	*säga*	**historia**
tillsätta	*tillägga*	**socker, salt, krydda**
lägga, lämna, ge	*göra*	**förslag**
tillbringa	*spendera*	**sommar, vinter**

Acceptable	**Not acceptable**	**Nouns**
uttala, uttrycka	*yttra*	**stöd**
ge	*tillsätta*	**smak**
tappa	*förlora*	**humör**
förstärka	*öka*	**tendens**
klara, genomgå	*passera*	**test**
klara, genomgå, avlägga	*passera*	**prov**
lansera	*introducera*	**teori, produkt**
avvisa, avfärda, förkasta	*avslå*	**tanke**
tillbringa, lägga	*spendera*	**tid**
ifrågasätta	*fråga*	**värde, rätt**
lägga, avge	*kasta*	**röst**
förklara	*deklarera*	**krig**
bana, jämna	*belägga*	**väg**
ligga	*vara*	**vecka**
lägga	*sätta, ställa*	**vikt**
söka	*leta*	**arbete, jobb**
fylla	*vända*	**år**

TRITA-CSC-E 2009:124
ISRN-KTH/CSC/E--09/124--SE
ISSN-1653-5715

www.kth.se