Automatic identification of nocuous ambiguity

Journal Article

Res Lang Comput (2008) 6:355–374
DOI 10.1007/s11168-008-9058-2

Automatic Identification of Nocuous Ambiguity

Alistair Willis · Francis Chantree · Anne De Roeck

Published online: 30 December 2008

Abstract We present the concept of *nocuous ambiguity*, which occurs when text is interpreted differently by different readers. In contrast, text exhibits innocuous ambiguity if different readers interpret it in the same way, even though structural or semantic analyses suggest that multiple interpretations may be possible. We collect multiple human judgements of a set of English phrases obtained from requirements documents. We focus on coordination ambiguity and show that across a group of judges there may be wide variation in what is perceived to be the correct interpretation. We develop the concept of an *ambiguity threshold*, which expresses the amount of variation between judgements that can be tolerated. We then develop and evaluate a heuristically based method of automatically predicting which sentences may be misunderstood for a given ambiguity threshold.

Keywords Nocuous ambiguity · Ambiguity notification · Ambiguity threshold · Coordination ambiguity

1 Introduction

Ambiguity is prevalent in natural language. However, when presented with an ambiguous sentence, it may not always be possible to determine which interpretation of the sentence is the one that the author intended. Consider the sentences (1a) and (1b).

A. Willis (✉) · F. Chantree · A. De Roeck
Department of Computing, The Open University, Milton Keynes, UK
e-mail: a.g.willis@open.ac.uk

F. Chantree
e-mail: f.j.chantree@open.ac.uk

A. De Roeck
e-mail: a.deroeck@open.ac.uk

(1) a. *They ate green beans and sausages*.
b. *They wore black hats and boots*.

Both sentences are generally considered to be ambiguous because of the different possible attachments of the adjectival modifier to the nouns. Most readers do not seem troubled by the ambiguity in (1a) and interpret the sentence in the same way; it is only the beans that are green, not the sausages. The preferred interpretation of the sentence turns on a question of meaning (Wasow et al. 2003); it is much more likely for beans than sausages to be green. Example (1b) appears to have the same surface syntactic structure as (1a), but it is much less clear how to interpret it, and whether the adjective *black* describes both hats and boots, or only the hats.

Situations where it is unclear how the text was intended to be interpreted can potentially lead to miscommunications. For example, sentence (2) is taken from a requirements document.

(2) *You must list all assumptions and dependencies that are of importance.*

The modifier *that are of importance* can apply either to the conjunction *assumptions and dependencies* or just to *dependencies*. If the requirements engineer intended that all assumptions should be listed, while the implementer interprets the requirement to mean that only the assumptions of importance should be listed, the system will not be correctly implemented. Automatic disambiguation does not help in such cases, where it is important to clarify what is intended. We say that sentences like (1b) and (2) exhibit *nocuous ambiguity* because they run a high risk of being interpreted differently by different people. Sentences like (1a) exhibit *innocuous ambiguity*: although (1a) can be given different syntactic analyses, speakers of English tend to interpret the sentence the same way in practice.

We are interested in whether nocuous ambiguity can be automatically identified. Central to modelling this concept is to see ambiguity not as a property of the text itself, but as a phenomenon which affects groups of interpreters of that text. We collect sentences from the domain of requirements engineering which can be given multiple syntactic analyses due to coordination ambiguity, and obtain sets of judgements on their interpretation. We develop a set of heuristics to predict whether an ambiguity is nocuous or innocuous, and then consider whether these heuristics are able to improve on baseline performances.

We envisage a practical use for this work. One approach to dealing with nocuous ambiguity in texts such as requirements documents may be to identify and highlight those sentences which different readers are likely to interpret differently. As Bernth (1998) notes, automatic disambiguation is inappropriate for highly ambiguous sentences; rather, users should be informed about how the text is ambiguous, and given the opportunity to rephrase it. Our approach has two added advantages. First, we do not attempt to highlight all sentences which contain structural ambiguities, but only identify those which display nocuous ambiguity. Second, we implement the model using ambiguity tolerance thresholds, so that the user can determine the degree of nocuity that the system should tolerate.

2 Modelling Nocuous Ambiguity

We are interested in discovering the factors which contribute to the preference for particular readings, and investigating whether the distinction between nocuous and innocuous ambiguity can be implemented in a computational model. We start by considering sentences that can in principle receive several different interpretations, such as (2) and (3).

(2) *You must list all assumptions and dependencies that are of importance.*
(3) *Every project is allocated a manager.*

Sentence (2) has multiple scoping options for quantifiers and modifiers, and seems to offer several likely candidates for interpretation. In contrast, (3) can theoretically be associated with different meanings depending on quantifier scope, but in practice will be interpreted in the same way by different people.

Clearly, in order to model the way a group of people might interpret the sentence, we need to use their individual interpretations. In practice, we do this by associating each sentence with a collection of human judgements, whose distribution is then used to classify the ambiguity as either nocuous or innocuous. The ambiguity threshold is an important concept in how we model the distinction between nocuous and innocuous ambiguity. We describe some related work before elaborating upon our empirical framework.

2.1 Related Work

The idea that some ambiguous expressions are more likely than others to be misunderstood, or that speakers are likely to prefer some interpretations over others, is well established. van Deemter's (2004) notion of *vicious ambiguity* describes the situation where speakers do not strongly prefer a single interpretation of a sentence over (all) other interpretations. van Rooy (2004) discusses similar issues, defining ambiguity as the situation where at least two interpretations of a sentence are equally relevant, or useful (Sperber and Wilson 1982) to the interpreter. Further, where puns and some other rhetorical techniques are used, there may deliberately be no single intended meaning (Poesio 1996).

There is a large body of work on syntactic disambiguation (Agarwal and Boggess 1992; Okumura and Muraki 1994; Resnik 1999). This literature is of limited relevance to our work: we aim to identify cases where ambiguity may lead to misunderstanding, not to resolve ambiguities. In machine translation, Emele and Dorna (1998) demonstrate how ambiguity can be maintained when translating between languages. Shemtov (1997) and Knight and Langkilde (2000) have argued it can be more efficient to generate fluent but possibly ambiguous text, rather than to attempt to eradicate all possible ambiguity. The argument that users should be notified about ambiguous language was previously suggested by the EasyEnglish project (Bernth 1998). In IBM's work on preprocessing technical manuals for machine translation, Bernth noted that systems could not be relied upon to automatically determine the correct interpretation. Such systems should notify the user that a construction could be ambiguous, and present a set of alternatives.

Our practical goal is to provide a system to assist writers of requirements documents by alerting them to nocuous ambiguity subject to some tolerance level. We are motivated by the potential costs of misunderstandings occurring at the requirements stage of the software lifecycle, where most work is done in natural language (Boehm 1981). In situations where an incorrect interpretation of text may lead to heavy costs, it is often more advisable to identify the potential misunderstanding, rather than attempting to disambiguate (Kamsties et al. 2001; Berry et al. 2000; Chantree et al. 2006). In the same context, Ceccato et al. (2004) introduce the notion of ambiguity thresholds in trying to measure levels of lexical ambiguity.

Our contribution differs from previous work in several ways. We treat ambiguity as a property of the relationship between a text and a group of interpreters, rather than as a property of the text itself (as is the case with syntactic disambiguation), or of its relation with a single interpreter. We use thresholds as a flexible means of distinguishing nocuous from innocuous ambiguity depending on the distribution of judgements and the degree of ambiguity that we are willing to tolerate. We follow an approach that emphasises notification of nocuous ambiguity to the user, who retains control over whether to preserve it or not. Unlike the EasyEnglish project, we do not seek to offer alternative interpretations.

3 Empirical Framework

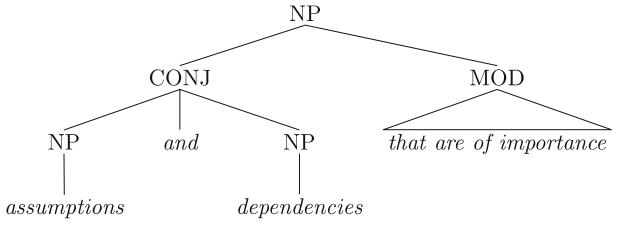
We have collected a corpus of requirements documents drawn from a variety of application domains, including engineering, health care and telecommunications. The documents were not filtered by geographic or cultural origin, nor were alterations made to perceived errors in the English.

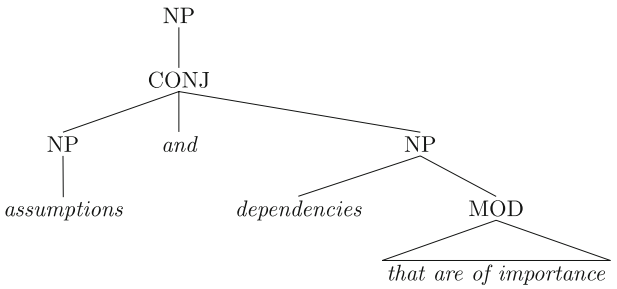
3.1 Coordination Ambiguity

We have focussed on the ambiguity of modifier attachment to coordinated phrases. The behaviour of these structures in technical documents has been discussed by Bernth (1998) and Ceccato et al. (2004). This is a common form of ambiguity in our document collection, and is marked by multiple possible syntactic structures for a phrase or sentence. For example, consider the phrase (4), taken from the sentence (2).

(4) *assumptions and dependencies that are of importance*

A phrase like (4) is generally thought to have two (syntactic) analyses due to coordination ambiguity:





The parse tree (5) represents an interpretation of the phrase where both *assumptions* and *dependencies* are modified by *that are of importance*, while the tree (6) represents an interpretation where only *dependencies* is modified. We say that (5) displays *high attachment* of the modifier, and that (6) displays *low attachment*. In general, we illustrate the ambiguity as in (7),

(7) [*assumptions and* (*dependencies*)] *that are of importance*

where the wide parentheses indicate the high attachment reading and the narrow parentheses indicate the low attachment reading. The modifier is underlined. We call *assumptions* and *dependencies* the *potential conjuncts*.

Coordination ambiguity is common in English, with different modifiers applying to different parts of speech.

(8) a. *for* [*use and* (*evaluation*)] *by others*
b. [*enter and* (*verify*)] *the data*
c. [*security and* (*privacy*)] *requirements*
d. *definition of electrical* [(*characteristics*) *and interfaces*]

For example, (8a) shows ambiguous attachment of a prepositional phrase to nouns, (8b) shows ambiguous NP attachment to verbs and (8c) shows ambiguity between possible conjoined NPs. The modifier can also appear before the potential conjuncts, as in (8d), which shows an adjectival modification.

3.2 Building a Dataset

From a set of requirements documents, we identified a set of 138 candidate phrases or sentences which could be assigned multiple syntactic structures on the basis of coordination ambiguity, with conjuncts connected by *and*, *or* and *and/or*.[1] By far the most common conjunction was between nouns (85.5%); that of verbs was comparatively infrequent (13.8%), and that of adjectives very rare (0.7%; only 1 instance in the set of 138 phrases). Nearly half of the cases arose as a result of nominal modifiers affecting possible readings of compound nouns (46.4%) although adjectival and prepositional attachment also accounted for a significant number (23.2% and 15.9%, respectively). From these sentences, we derived a dataset from which to elicit human judgements. We sought to avoid overloading judges with sentence length or complexity and simplified

[1] *and/or* is frequently used as a conjunction in requirements documents.

sentence parts that we felt were clearly not essential to the interpretation, such as preambles, trailing clauses, and proprietary names that were not head nouns. In the examples presented to the judges, the coordinations and modifiers were highlighted as shown in Sect. 3.1. Of these, approximately one third were verbatim sentences (such as (2)), with the remaining either phrases taken in isolation (such as those in (8)) or sentences with simplifications represented by conventional typographic notation (such as parentheses, dots etc.), as shown in (9).

(9) (*it*) *will be* [*implemented and* (*executed*)] *on the…platform*

In collecting judgements, the number of judges must be large enough to minimise the effect of noise introduced by rogue judgements [2] (Keren 1992). The number chosen should establish a basic level of confidence in the data; the lower bound is achieved if the judges can be shown to give informed (i.e. better than random) judgements. We require enough judges so that we are able to reject the null hypothesis, that the judges are responding randomly in the questionnaire. On the basis of pilot studies, we estimated that we should tolerate up to two rogue judges. In collecting the dataset, we were able to use 17 judges altogether; following Umarji (1962), we determined that this is a large enough number of judges to be able to reject the null hypothesis at a confidence level of 95%. These 17 judges were all familiar with software requirements, and had a good command of English. All of the judges worked in computing, and were lecturers, research students or software developers. All were living in the UK at the time of the experiment and were fluent speakers, with 13 being native speakers.

The sample phrases containing potential ambiguity were split into four questionnaires, which were presented to the judges in managed face to face sessions (four separate sessions, with one questionnaire used per session). The judges had the task explained and illustrated in detail before being asked to record their interpretations of the actual questionnaire examples. The order of examples was randomised, so that each judge received the examples in a different order. Judges were then asked to record their interpretations, either by circling or highlighting the part of the co-ordination affected by the modifier, or by indicating that they experienced the item as ambiguous (that is, that they did not feel able to choose between the two possible interpretations). Each judge gave one of three judgements for each sentence: high attachment of the modifier, low attachment of the modifier, or ambiguous, which we call "acknowledged ambiguity" for that judge. It is possible that this methodology introduced some response bias and noise in the data. Specifically, the task may have been insufficiently clear for a non-linguistic audience, the emphasis placed on the presence of ambiguity may have encouraged participants towards an acknowledged ambiguity judgement, and the degree of English proficiency may have had an effect.

The pool of judges contained five people with a background in computational linguistics. Numbers are small, but a check of their judgements revealed no evidence that they are more likely than others to return acknowledged ambiguity judgements. The check did suggest there may be an inverse correlation between the English fluency of a judge and the number of acknowledged ambiguity judgements returned, but we

[2] Rogue judgements are errors made by judges through carelessness or by accident, rather than judgements that reflect a genuine difference of opinion.

Table 1 Judgement counts for phrases (8c) and (8d)

	Judgements		
	HA	LA	A
(8c) [*security and* (*privacy*)] *requirements*	12	1	4
(8d) *electrical* [(*characteristics*) *and interfaces*]	4	4	9

deemed this an acceptable bias given that the domain of requirements and software engineering involves a diverse, international professional community. Finally, the possible overestimate of acknowledged ambiguity in the data should have no practical adverse effects; at worst, it would lead to more cases of nocuity being flagged up and the effect can to an extent be managed by varying the ambiguity threshold (see Sect. 3.3).

Each of the 138 candidate phrases has 17 judgements associated with it (very occasionally fewer, if a judgement was missed, although less than 1% of judgements were absent, and these omissions were spread across candidate phrases. We have ignored these as they represent such a small percentage). For example, the judgements for the phrases (8c) and (8d) are shown in Table 1, where "HA", "LA" and "A" represent judgements of "high attachment", "low attachment" and "ambiguous", respectively. Phrase (8c) was judged mainly to have high attachment of the modifier, while phrase (8d) was mainly judged to be ambiguous.

In practice, it is very unusual for judges to divide between high attachment and low attachment for a given phrase. Where there is disagreement, the judgements tend to divide between one of the non-ambiguous options, and assigning an ambiguous judgement. In our dataset, where judges disagree with the preferred non-ambiguous judgement, an average of 18.7% of dissenters assign the minority non-ambiguous judgement. The other 81.3% assign the "ambiguous" judgement. In only four cases did the numbers of judgements in favour of the minority non-ambiguous option exceed those of the "ambiguous" option. This indicates that readers probably did exercise caution, expressing any doubts they had by acknowledging the ambiguity.

3.3 Nocuity and Thresholds

We use the idea of an ambiguity threshold to represent our tolerance of ambiguity. The threshold represents how much agreement we require from the judges on a particular interpretation before declaring that the ambiguity is innocuous. For a high threshold, an ambiguity is classified as innocuous only if there is a clear agreement between the judges on a particular interpretation.

Given an ambiguous phrase and a set of judgements, the *certainty* of an interpretation is simply the percentage of judgements of that interpretation. For example, consider the phrase (8c) in Table 1. The certainty of the high attachment interpretation is $12/17 \approx 71\%$, and the certainty of the low attachment interpretation is $1/17 \approx 5.9\%$.

We define nocuous ambiguity in terms of certainty as:

Definition Given an ambiguous phrase or sentence, S, a collection of judgements of the correct interpretation of that sentence and an ambiguity threshold T (where $0 \leq T \leq 100\%$):

if there is at least one non-ambiguous interpretation of S which has a certainty greater than T, then S exhibits *innocuous ambiguity* at threshold T. Otherwise, S exhibits *nocuous ambiguity* at threshold T.

So at an ambiguity threshold of 80%, the phrase (8c) exhibits nocuous ambiguity; neither the high attachment interpretation (with a certainty of 71%), nor the low attachment interpretation (with a certainty of 5.9%) is higher than this threshold. However, at an ambiguity threshold of 50%, the sentence exhibits innocuous ambiguity; the certainty of the high attachment interpretation is above the threshold. However, the phrase (8d) displays nocuous ambiguity at a threshold of 50%. About 53% of the judgements of (8d) were acknowledged ambiguity, but as neither of the non-ambiguous judgements had a certainty greater than 50%, we still say that the sentence exhibits nocuous ambiguity at this threshold.

The relationship between the ambiguity threshold and the classification of ambiguity as either nocuous or innocuous is illustrated in Fig. 1. At low thresholds, very few phrases exhibit nocuous ambiguity, because a small number of non-ambiguous judgements is enough to raise the proportion over the threshold. However, for high thresholds, almost all ambiguity is classified as nocuous. Unless most judges agree on the same interpretation, no one interpretation has a certainty that is as high as the chosen threshold.

The value of an ambiguity threshold to highlight potential misunderstandings to a user has been discussed by Chantree et al. (2006) and Ceccato et al. (2004), both of whom consider the application to requirements documents. In particular, Chantree et al. argue that allowing a user to set the ambiguity threshold allows systems that allow little ambiguity in documents relating to safety critical systems (for example), while allowing more tolerance of ambiguity in preliminary discussion documents.

Note that our view of the ambiguity threshold contrasts with that proposed by Ceccato et al., who propose a measurement of the ambiguity of a sentence based upon the number of different senses of its constituent words. This measure of ambiguity, in terms of the properties of the text, contrasts with our own view, which considers ambiguity in terms of the distribution of multiple human judgements.

Because the threshold allows varying degrees of sensitivity to ambiguity, we believe that the potentially high level of acknowledged ambiguity in the data set can be accommodated by setting a lower threshold. We have already noted that response bias may

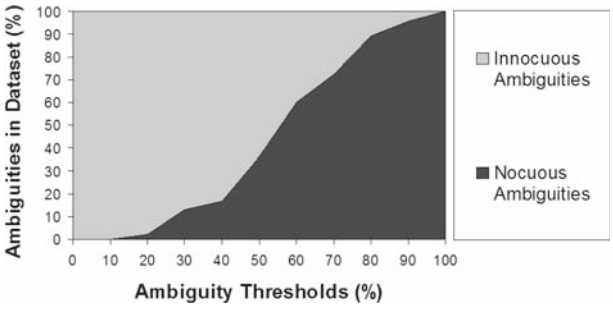


Fig. 1 Proportions of interpretations at different ambiguity thresholds

increase the number of acknowledged ambiguity judgements. Also, the data is stripped of context (including syntactic context) when presented in the questionnaires. It is possible that this too gives rise to an increased number of acknowledged ambiguity verdicts. By using a low ambiguity threshold, sentences can be considered to have a preferred interpretation, even if the judgements suggest a lower certainty than would be expected with a complete context.

4 Heuristics to Predict Nocuity

Section 3.3 defined nocuous ambiguity relative to some ambiguity threshold based on a set of human judgements. We now consider whether it is possible to predict whether a particular sentence displays nocuous ambiguity.

We propose a set of heuristics to apply to sentences which display coordination ambiguity. Each of the individual heuristics attempts to identify an aspect of the sentence that may lead an interpreter to prefer either a high attachment or low attachment interpretation. If the heuristics together predict a strong preference for a particular interpretation, then the ambiguity is predicted to be innocuous, otherwise the ambiguity is predicted to be nocuous (we consider how they should be combined to predict overall nocuity in Sect. 5). Many of the heuristics are based upon word distributions in English (for example, whether two words are frequently coordinated). In these cases, the word distributions are obtained from the British National Corpus (BNC) [3] via the Sketch Engine (Kilgarriff et al. 2004). Because our dataset is drawn entirely from the domain of requirements documents, it is possible that some word distribution information obtained from the generic BNC may not accurately reflect the use of the same words among requirements documents. However, work on the REVERE project (Sawyer et al. 2002) has demonstrated that the BNC can be successfully used as the representative corpus for applications involving requirements documents.

We show the performance of the individual heuristics in terms of their ability to select those sentences whose majority judgement is the interpretation identified by that heuristic. For example, the coordination matching heuristic (Sect. 4.1) identifies high attachment; its performance is given in terms of its ability to select those sentences which were judged high attachment more than either low attachment or acknowledged ambiguity.

We have selected heuristics which give high precision at the expense of recall. A particular heuristic will usually select correct phrases, but the number selected by that heuristic may be small. We believe that although each individual heuristic has low recall, combining several heuristics will improve the overall recall. As the individual heuristics are selected for high precision, the combination of heuristics will then cover enough of the test phrases to predict nocuity with greater accuracy. We use an f-measure (van Rijsbergen 1979) which uses $\beta = 0.25$ to give a weighting towards precision:

$$f = \frac{(1+\beta) * precision * recall}{\beta^2 * precision + recall}$$

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

The results graphs show a baseline precision at 54.3%. This is the precision obtained by assuming that the consensus judgement (the judgement with the highest certainty) of all phrases is acknowledged ambiguity. This is a higher baseline than assuming a consensus judgement of either high attachment or low attachment.

4.1 Coordination Matching

This heuristic proposes that if the head words of the two conjuncts are frequently coordinated in the language, then that coordination forms a single syntactic unit. The particular sentence should therefore have high attachment of the modifier.

To calculate the value returned by this heuristic, the word sketch facility of the Sketch Engine is used, which generates frequency lists of words that are conjoined with *and* or *or* in the BNC. For a given sentence containing a modified conjunction ambiguity, the heuristic's value is given by the ranked frequency of coordination between the head words of the two conjuncts. For example, consider again the phrase (8c):

(8c) [*security and* (*privacy*)] *requirements*

security is the 8th most commonly coordinated word with *privacy*, so the value returned by the heuristic is 8. In fact, the ranking must be found for both head words as the frequency of coordination can be different for the two words. So in this case, although *security* is the 8th most commonly coordinated word with *privacy*, *privacy* is only the 19th most commonly coordinated word with *security*. In such cases we use the higher of the two rankings, which in this case is 8. Phrase (8c) was most commonly judged to have high attachment, so for this phrase, the heuristic gives a true positive result for any cutoff that requires a ranking of 8 or higher.

By using rankings, rather than absolute frequency counts, we hope to prevent this heuristic indicating a disproportionate preference for high attachment when applied to frequently occurring words. If the two head words in the conjuncts both appear very frequently independently, but are rarely conjoined, the heuristic will return a low preference for high attachment, as the ranked frequency will be low.

The performance of this heuristic, using different ranking cutoffs in multiples of 5, is shown in Fig. 2. Precision in excess of 21% points above the baseline and 40% recall can be achieved indicating that, even on its own, this heuristic is a useful predictor of high attachment readings.

4.2 Distributional Similarity

This heuristic, based on a proposal Kilgarriff (2003), proposes that if the head words of the two conjuncts have a high distributional similarity, then the coordinated phrase forms a syntactic unit and the particular sentence should therefore have high attachment of the modifier. The distributional similarity of two words is a measure of how often those words are found in the same contexts. For instance, *good* and *bad* have strong distributional similarity, although though their meanings are opposite.

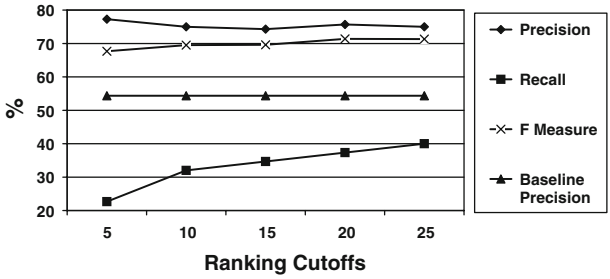


Fig. 2 Coordination matching heuristic predicting high attachment

The Sketch Engine contains a distributional thesaurus in the tradition of Spärck-Jones (1986) and Grefenstette (1994). For this heuristic, we again use the ranking given by the Sketch Engine for the head words of the two potential conjuncts, this time using the distributional thesaurus to give a measure of distributional similarity (Lin 1998). As with the coordination matching heuristic, we perform this procedure for each head word of a coordination, using the higher of the two rankings as our metric.

For example, consider again (8d).

(8d) *electrical* [(*characteristics*) *and interfaces*]

Of the 17 judges, 9 judged this to be ambiguous, and so the heuristic should be expected not to predict the high attachment interpretation. In fact, the distributional thesaurus finds no matches between *characteristic* and *interface*, and so the heuristic yields a true negative result.

The results for the distributional similarity heuristic are shown in Fig. 3. Experimentation has shown us that the predictive power of distributional similarity decreases in a log-linear manner the more matches are considered, so we choose cutoffs accordingly. As can be seen, precision in excess of 20% points above the baseline can be achieved, though recall at this level reaches only 20%.

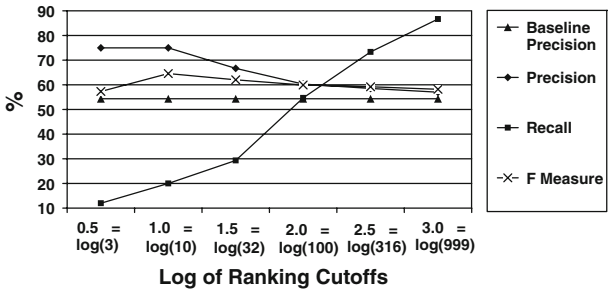


Fig. 3 Distributional similarity heuristic predicting high attachment

4.3 Morphology

This heuristic predicts high attachment interpretations, and is based upon Okumura and Muraki's (1994) notion of using syntactic parallelism as a method of disambiguating coordinations. We hypothesise that if the head words of the potential conjuncts share a similar morphology, then they form a syntactic unit, resulting in preference for high attachment. The inflectional morphology of English consists largely of suffixes such as *-ed* to indicate past tense, *-ing* to indicate progressive action, and *-s* to indicate plurals. The derivational morphology of English is more complex but suffixes, such as *-ation* and *-able*, are also very common.

The value returned by this heuristic is the number of common trailing characters of the head words of the potential conjuncts. Consider sentence (10).

(10) *It cannot function with the proper [(installation) and configuration].*

The last five trailing characters of *installation* match those of *configuration*, so the heuristic gives a positive result for this coordination up to a cutoff of 5. Two survey judges judged this coordination to be ambiguous, thirteen judged it to have high attachment, nobody judged it to have low attachment and two judges entered no response. Therefore the heuristic gives a true positive result for any cutoff up to 5.

The results that we obtain using different ratios are shown in Fig. 4. We considered cutoffs from one character (capturing most plurals but also a lot of noise) to six (by which point, capturing morphology has become vanishingly small). Maximum precision is achieved at a cutoff of 5, where it is more than 45% points above the baseline, although recall is only 2.7% at this cutoff. Also, the number of cases with five common trailing characters is small (there are only two such phrases in the dataset), although three and four trailing characters are more common. This suggests that the heuristic has only a limited contribution to make to the predictive power of the combined heuristics, but it appears to be a reliable one.

4.4 Collocation Frequency

This heuristic proposes that if the modifier is collocated in the language much more frequently with the head word of the nearer potential conjunct than it is collocated

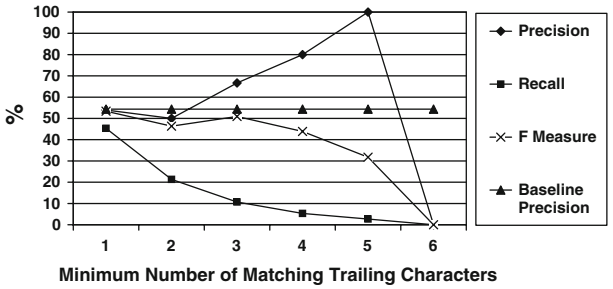


Fig. 4 Morphology heuristic predicting high attachment

with the head word of the further potential conjunct, then the modifier and the nearer conjunct form a single syntactic unit. The sentence should therefore display low attachment of the modifier. Collocation frequencies have already been successfully used in disambiguation tasks (Nakov and Hearst 2005; Rus et al. 2002).

To calculate the value returned by this heuristic, the collocation frequency functions of the Sketch Engine are used. The value returned is the ratio of the collocation frequency with the nearest head word over the collocation frequency with the further head word. For example, consider phrase (11) from the dataset.

(11) *project* [(*manager*) *and designer*]

project has a collocation frequency of 29.55 with *manager* in the BNC (obtained via the Sketch Engine), but it has no collocations with *designer*. Therefore the heuristic yields a positive result for (11) (we assume that division by zero is a true result for all cutoffs). Of the 17 judges, (11) was judged to be ambiguous by 5, to have high attachment by 4 and low attachment by 8. Therefore, this heuristic gives a true positive result for predicting low attachment readings.

The results we obtain using different collocation frequency ratios are shown in Fig. 5, which shows the precision and recall of using different minimum values of the ratio to predict innocuous ambiguity. Precision of 6.5% points above the baseline can be achieved, with recall of 18.7%. This performance is modest, but as this heuristic is the only one that successfully predicts low attachment interpretations, it is a vital contribution to the combined model.

Note, however, that the possibility of finding collocations for many examples of coordination ambiguity will be small due to combinatorial factors.

(12) *facilitate the* [*scheduling and* (*performing*)] *of works*

For example, in (12), no collocations are found either between either *scheduling* and *works* or *performing* and *works*. It could be that there is low likelihood of ever finding *works* associated with either *scheduling* or *performing*. However, it is probably more likely because the preposition of must be considered as well. The possibility of finding the phrase of works as a modifying collocation is much less likely.

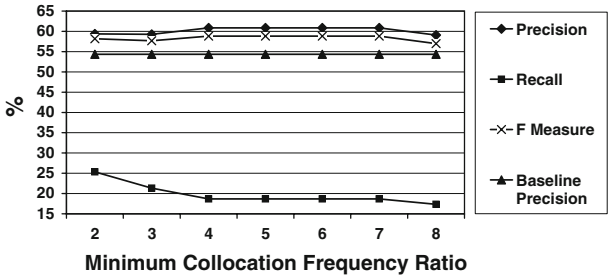


Fig. 5 Collocation frequency heuristic predicting low attachment

4.5 Other Heuristics Used

This section briefly describes three further heuristics. These showed little predictive power individually, but when all heuristics were used in combination (Sect. 5), better results were obtained when these were included.

Following Resnik (1999), we propose a *noun number agreement* heuristic, which predicts a high attachment interpretation if the head nouns agree in number. Also, a *mass/count* heuristic predicts a high attachment interpretation if the head nouns of the potential conjuncts are either both mass nouns or both count nouns. The noun number agreement and mass/count heuristics apply only where the potential conjuncts are both nouns or NPs.

Finally, Schepman and Rodway's (2000) investigations into the effect of prosodic boundaries in (spoken) coordination disambiguation suggest that phrase length can affect interpretation. We use a *phrase length difference* heuristic which predicts a low attachment interpretation. The value returned by the heuristic is difference between the number of words in each of the potentially conjoined phrases, where a high value (so greater difference in phrase length) is taken as evidence for low attachment.

5 Combining the Heuristics

For the applications that we have discussed, such as ambiguity notification, we are interested only in predicting whether a given sentence exhibits nocuous ambiguity and so runs the risk of being misinterpreted; we do not attempt to identify an intended meaning. To identify nocuous ambiguity, we need to consider the certainty of the consensus judgement, rather than what that consensus judgement might be. However, each of the heuristics described in Sect. 4 only indicates a preference for high or low attachment. Individually, they do not predict whether a given sentence displays nocuous ambiguity at a given ambiguity threshold.

In this section, we discuss how the individual heuristics can be combined to predict whether a particular sentence displays nocuous ambiguity. We describe how a linear logistic regression model was used to combine the individual heuristics into a predictor for nocuous ambiguity, and compare the performance of the trained model against a set of baselines.

5.1 Logistic Regression Model

Given a particular sentence and an ambiguity threshold, we wish to estimate whether the sentence displays nocuous ambiguity at that threshold or not. We postulate that the likelihood of the sentence displaying nocuous ambiguity can be predicted by using the values generated by a suitable combination of the heuristics described in Sect. 4.

To find a suitable way of combining the heuristics, we have used a linear logistic regression model (Hosmer and Lemeshow 2000) to model the relationship between the values of the heuristics found for each sentence, and the probability of the sentence

displaying nocuous ambiguity. In this way, the model can be used to classify phrases as either nocuous or innocuous. This is a typical regression analysis; the parameters of the model are estimated using our existing data, and for new cases, we can estimate the probability that the sentence is also nocuous.

In logistic regression, the behaviour of a dependent variable, Y, is modelled by a vector of n explanatory variables, $\langle x_1, x_2, \ldots, x_n \rangle$. Y must take one of two values (true or false), while each of the x_i is drawn from a numerical domain. The explanatory variables need not be drawn from the same domain; some variables might take integer values, while others take real numbers. The regression model then assumes that Y is related to the x_i by the equation:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

where p is the probability that Y is true, given the particular values of $x_1, \ldots x_n$. The β_i can be obtained by training the model with existing data (Sect. 5.2).

In our case, the two possible values of Y represent either nocuous or innocuous ambiguity for a sentence, and the values of the explanatory variables are the values returned by the seven heuristics described in Sect. 4 for that sentence. Of course, whether a sentence displays nocuous ambiguity or not depends upon the chosen ambiguity threshold; a sentence that displays nocuous ambiguity at a high threshold may be innocuous at a lower threshold. The coefficients in the regression equation must be found for the particular ambiguity threshold. So given an ambiguity threshold T, we require the equation:

$$\log\left(\frac{p}{1-p}\right) = \beta_0^T + \beta_1^T x_1 + \beta_2^T x_2 + \cdots + \beta_7^T x_7 \quad (1)$$

where the β_i^T are the coefficients for the particular ambiguity threshold T. For a particular sentence, the explanatory variables $x_1, \ldots x_7$ are the values generated by the heuristics for that sentence, and p is the probability that the sentence displays innocuous ambiguity at the threshold T. We then classify a sentence as displaying innocuous ambiguity at threshold T if Eq. 1 gives a value of p which is greater than T. That is, the sentence is classified as innocuous if Eq. 1 assigns it a probability p of being innocuous, and p is itself above the ambiguity threshold.

5.2 Training the Regression Model

To find the coefficients β_i^T in Eq. 1, we use the LogitBoost algorithm (Friedman et al. 2000) which is implemented within the WEKA machine learning algorithms (Witten and Frank 2005). The LogitBoost algorithm is designed to maximise the accuracy of the prediction.

The training data is obtained from the dataset of sentences and the associated judgements about whether the sentence displays high attachment, low attachment, or is an acknowledged ambiguity (Sect. 3.2). The training values of the explanatory variables $x_1, \ldots x_7$ are the values obtained by applying each of the heuristics to

the sentences. The value of the dependent variable is set at 1 for sentences which are judged to display innocuous ambiguity, and 0 for sentences which are judged to display nocuous ambiguity (an ambiguity is innocuous if the certainty of at least one non-ambiguous interpretation is greater than the ambiguity threshold). Because nocuous ambiguity is defined in terms of the ambiguity threshold, the model generated by a particular training instance is specific to that threshold. For the evaluation, we obtained sets of coefficients β_i^T for values of T at increments of 5% (that is, $T = 0, T = 5\%, T = 10\% \ldots T = 100\%$).

A particular issue in training the regression model is the relationship between the explanatory variables and the dependent variable. For a trained model, higher values for the explanatory variables will increase the predicted probability that the ambiguity is innocuous. This behaviour is generally desirable. For example, if a sentence has high values for those heuristics which predict high attachment (such as the coordination matching heuristic), then it may be reasonable to predict innocuous ambiguity; there is evidence for a preferred interpretation. However, if a sentence has high values for both the heuristics predicting high attachment and the heuristics predicting low attachment, there may be an apparent conflict. High values for both the high attachment heuristics and the low attachment heuristics should presumably be interpreted as evidence for nocuous ambiguity, but may be interpreted by the regression model as evidence for innocuous ambiguity.

In practice, however, this situation seems to be extremely uncommon. There is only one example of the 138 sentences in the dataset which scores highly for both the high attachment and low attachment heuristics (this sentence was judged to be low attachment with a confidence of 59%). In this sense, the behaviour of the heuristics appears to be similar to that of the human judges. As noted in Sect. 3.2, the judges tended to divide between one of the non-ambiguous interpretations and acknowledged ambiguity, rather than dividing between high attachment and low attachment. This is also the behaviour observed with the heuristics.

5.3 Evaluation

To avoid the problem of overfitting, we have used the 10-fold cross-validation facilities offered by the WEKA machine learning package. Ten-fold cross-validation is recognised as an appropriate method for datasets of the size of ours (Weiss and Kulikowski 1991).

To evaluate how well the combined heuristics can be used to predict whether a sentence is nocuous or innocuous, we require a baseline measure to compare against. A first attempt might be either to compare against a strategy of predicting all ambiguity to be nocuous, or all ambiguity to be innocuous. In fact, neither of these strategies alone provides a suitable baseline. As we have already seen, for low ambiguity thresholds, most ambiguity is innocuous, while for high ambiguity thresholds, most ambiguity is nocuous (Sect. 2). Therefore, at low thresholds, a good strategy will simply be to predict that all ambiguity is innocuous. Similarly, at high thresholds we can get good accuracy simply by predicting that all ambiguity is nocuous.

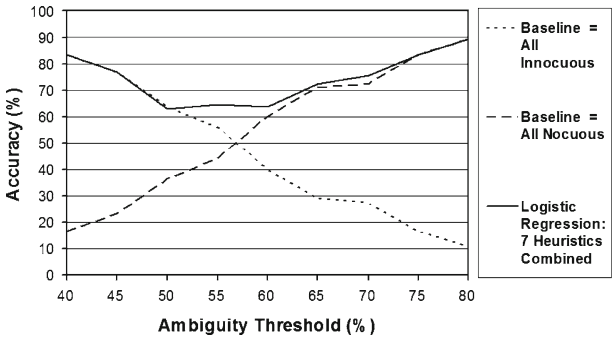


Fig. 6 Combined heuristics' accuracy

Therefore, the baseline accuracy measure is found by predicting either that all ambiguity is innocuous or that all ambiguity is nocuous, whichever is the more accurate predictor for that particular ambiguity threshold. For our dataset, the strategy of predicting all innocuous has higher accuracy for thresholds below 58%, while predicting all nocuous has higher accuracy over 58% (the baseline accuracy measures are shown in Fig. 6). Because a high baseline prediction accuracy is generally very hard to outperform (Manning and Schütze 1999), we do not expect to be able to outperform the baseline for very high or very low ambiguity thresholds. However, we would anticipate improvements at and around the 58% crossover point, where the baseline prediction accuracy is poorest.

5.4 Results

The performance of the combined heuristics, along with the baselines, is shown in Fig. 6, which covers the range of ambiguity thresholds where the predictor best outperforms the baseline strategies. As expected, the heuristics cannot outperform the conservative baselines at very high and very low thresholds. However, we do note some improvement where the baselines are at their weakest, in the range that is shown.

Figure 6 shows that the regression model can outperform the baselines for ambiguity thresholds between around 50% and 70%. The maximum improvement in performance is where the two baseline strategies intersect, where the accuracy of the model is 14% better than the baselines.

Between ambiguity thresholds of 60% and 75% the performance is less marked, with an average improvement in performance in this range of around 2.7% above the baseline, although the baselines are more accurate themselves in this range. Between the thresholds of 40% and 50%, there is no improvement in performance above the baseline. The improvements over the baselines for ambiguity thresholds greater than 60% are less conclusive, but do show that the combined heuristics have some ability at distinguishing nocuous from innocuous ambiguity at these levels.

The 50% ambiguity threshold, where the fitted model makes the best improvement on the baseline, may be appropriate for many applications. This threshold represents

a trade-off between identifying potentially dangerous ambiguous constructions, and tolerating less dangerous ones.

6 Conclusion

We have defined the phenomenon of *nocuous ambiguity*, which occurs when an ambiguous expression runs an unacceptable risk of being interpreted differently by different people. Importantly, any one reader may be unaware that others understand the expression differently, and therefore no individual can state whether a sentence exhibits nocuous ambiguity. By collecting judgements from multiple speakers, we have demonstrated that nocuous ambiguity does exist in requirements documents. We have argued that under these circumstances, we should not attempt to disambiguate such text automatically. Instead, it is preferable to notify the user of the potential misunderstanding. We have used thresholds to model the degree to which a user will tolerate misunderstandings, and have developed a model that trains heuristics to predict when nocuous ambiguity is likely to occur, given a particular threshold. The higher the threshold, the more consensus is required in order to state that an ambiguity is innocuous.

Although we believe that nocuity is a property of ambiguity in general, the heuristics we have proposed are specific to the coordination ambiguities we have investigated: different types of ambiguity, such as word sense ambiguity or PP attachment ambiguity, would of course require different heuristics. We set out a new evaluation regime to test our approach, using competing naïve baselines for nocuous and innocuous ambiguity. We showed that the heuristics improved at those ambiguity thresholds where the baselines performed poorly.

Our future work will address some of the questions raised by our current approach. Further steps are needed to reduce the effects of noise and bias in the collected judgements. Additional heuristics need to be developed to account for more types of ambiguity, and the extent to which the existing heuristics are affected by our use of the British National Corpus (rather than on text drawn from the same domain—in this case requirements documents) needs to be examined. In addition, our evaluation measures should take more account of the effect of absolute word frequencies when evaluating the performance of the heuristics, both for very frequently occurring words, and where data sparseness may affect performance. However, our current results do suggest that heuristics trained on a general corpus can be applied to a specialised domain.

References

Agarwal, R., & Boggess, L. (1992). A simple but useful approach to conjunct identification. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pp. 15–21.

Bernth, A. (1998). EasyEnglish: Addressing structural ambiguity. In *Machine Translation and the Information Soup: Proceedings of the Third Conference of the Association for Machine Translation in the Americas, Vol. 1529 of Lecture Notes in Artificial Intelligence* (pp. 164–173). Springer-Verlag.

Berry, D. M., Kamsties, E., & Krieger, M. M. (2000). From contract drafting to software specification: Linguistic sources of ambiguity. Technical Report, University of Waterloo, Waterloo, ON, Canada.

Boehm, B. W. (1981). *Software engineering economics*. Englewood Cliffs, NJ, USA: Prentice-Hall.
Ceccato, M., Kiyavitskaya, N., Zeni, N., Mich, L., & Berry, D. M. (2004). Ambiguity identification and measurement in natural language texts. Technical Report, University of Trento.
Chantree, F., Nuseibeh, B., de Roeck, A., & Willis, A. (2006). Identifying nocuous ambiguities in requirements specifications. In M. Glinz & R. Lutz (Eds.), *Proceedings of the 14th IEEE International Requirements Engineering conference*, pp. 59–68.
Emele, M. C., & Dorna, M. (1998). Ambiguity preserving machine translation using packed representations. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pp. 365–371.
Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *The Annals of Statistic, 38*(2), 337–374.
Grefenstette, G. (1994). *Explorations in automatic thesaurus discovery*. Kluwer Academic Publishers.
Hosmer, D. W., & Lemeshow, S. (2000). *Applied logistic regression. Wiley series in probability and statistics*. Wiley.
Kamsties, E., Berry, D. M., & Paech, B. (2001). Detecting ambiguities in requirements documents using inspections. In M. Lawford & D. L. Parnas (Eds.), *Proceedings of the First Workshop on Inspection in Software Engineering (WISE'01)*, pp. 68–80.
Keren, G. (1992). Improving decisions and judgments: The desirable versus the feasible. In G. Wright & F. Bolger (Eds.), *Expertise and decision support* (pp. 25–46). New York: Plenum Press.
Kilgarriff, A. (2003). Thesauruses for natural language processing. In C. Zong (Ed.), *Proceedings of NLP-KE*, Beijing, China, pp. 5–13.
Kilgarriff, A., Rychly, P., Smrz, P., & Tugwell, D. (2004). The sketch engine. In G. Williams & S. Vessier (Eds.), *Proceedings of the Eleventh European Association for Lexicography (EURALEX) International Congress*, pp. 105–116.
Knight, K., & Langkilde, I. (2000). Preserving ambiguities in generation via automata intersection. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence* (pp. 697–702). AAAI Press/The MIT Press.
Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pp. 768–774.
Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA, USA: MIT Press.
Nakov, P., & Hearst, M. (2005). Using the Web as an implicit training set: Application to structural ambiguity resolution. In *Proceedings of HLT-NAACL'05*, pp. 835–842.
Okumura, A., & Muraki, K. (1994). Symmetric pattern matching analysis for English coordinate structures. In *Proceedings of the 4th Conference on Applied Natural Language Processing*, pp. 41–46.
Poesio, M. (1996). Semantic ambiguity and perceived ambiguity. In K. van Deemter & S. Peters (Eds.), *Semantic ambiguity and underspecification* (pp. 159–201). Cambridge, England: Cambridge University Press.
Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research, 11*, 95–130.
Rus, V., Moldovan, D. I., & Bolohan, O. (2002). Bracketing compound nouns for logic form derivation. In *Proceedings of the FLAIRS 2002 Conference*, pp. 198–202.
Sawyer, P., Rayson, P., & Garside, R. (2002). REVERE: Support for requirements synthesis from documents. *Information Systems Frontiers, 4*(3), 343–353.
Schepman, A., & Rodway, P. (2000). Prosody and parsing in coordination structures. *The Quarterly Journal of Experimental Psychology: A, 53*(2), 377–396.
Shemtov, H. (1997). Ambiguity Management in Natural Language Generation. Ph.D. thesis, Stanford University, USA.
Spärck-Jones, K. (1986). *Synonymy and semantic classification*. Edinburgh University Press.
Sperber, D., & Wilson, D. (1982). Mutual knowledge and relevance in theories of comprehension. In N. Smith (Ed.), *Mutual knowledge*. London: Academic Press.
Umarji, R. (1962). *Probability and statistical methods*. Bombay: Asia Publishing House.
van Deemter, K. (2004). Towards a probabilistic version of bidirectional OT syntax and semantics. *Journal of Semantics, 21*(3), 251–280.
van Rijsbergen, C. J. (1979). *Information retrieval*. London, UK: Butterworths.

van Rooy, R. (2004). Relevance and bidirectional OT. In R. Blutner & H. Zeevat (Eds.), *Optimality theory and pragmatics* (pp. 173–210). Basingstoke, Hampshire, UK: Palgrave/Macmillan.

Wasow, T., Perfors, A., & Beaver, D. (2003). The puzzle of ambiguity. In O. Orgun & P. Sells (Eds.), *Morphology and the Web of grammar: Essays in memory of Steven G. Lapointe*. CSLI Publications.

Weiss, S. M., & Kulikowski, C. A. (1991). *Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). San Francisco: Morgan Kaufmann.