# A Methodology for Automatic Identification of Nocuous Ambiguity

**Hui Yang**[1] **Anne de Roeck**[1] **Alistair Willis**[1] **Bashar Nuseibeh**[1, 2]
[1]Department of Computing, The Open University
[2]Lero, University of Limerick
{h.yang, a.deroeck, a.g.willis, b.nuseibeh}@open.ac.uk

## Abstract

Nocuous ambiguity occurs when a linguistic expression is interpreted differently by different readers in a given context. We present an approach to automatically identify nocuous ambiguity that is likely to lead to misunderstandings among readers. Our model is built on a machine learning architecture. It learns from a set of heuristics each of which predicts a factor that may lead a reader to favor a particular interpretation. An ambiguity threshold indicates the extent to which ambiguity can be tolerated in the application domain. Collections of human judgments are used to train heuristics and set ambiguity thresholds, and for evaluation. We report results from applying the methodology to coordination and anaphora ambiguity. Results show that the method can identify nocuous ambiguity in text, and may be widened to cover further types of ambiguity. We discuss approaches to evaluation.

## 1 Introduction

Traditional accounts of ambiguity have generally assumed that each use of a linguistic expression has a unique intended interpretation in context, and attempted to develop a model to determine it (Nakov and Hearst, 2005; Brill and Resnik, 1994). However, disambiguation is not always appropriate or even desirable (Poesio and Artstein, 2008). Ambiguous text may be interpreted differently by different readers, with no consensus about which reading is the intended one. Attempting to assign a preferred interpretation may therefore be inappropriate. Misunderstandings among readers do occur and may have undesirable consequences. In requirements engineering processes, for example, this results in costly implementation errors (Boyd et al., 2005).

Nonetheless, most text does not lead to significant misinterpretation. Our research aims to establish a model that estimates how likely an ambiguity is to lead to misunderstandings. Our previous work on nocuous ambiguity (Chantree et al., 2006; Willis et al., 2008) cast ambiguity not as a property of a text, but as a property of text in relation to a set of stakeholders. We drew on human judgments - interpretations held by a group of readers of a text – to establish criteria for judging the presence of nocuous ambiguity. An ambiguity is *innocuous* if it is read in the same way by different people, and *nocuous* otherwise. The model was tested on co-ordination ambiguity only.

In this paper, we implement, refine and extend the model. We investigate two typical ambiguity types arising from coordination and anaphora. We extend the previous work (Willis et al., 2008) with additional heuristics, and refine the concept of ambiguity threshold. We experiment with alternative machine learning algorithms to find optimal ways of combining the output of the heuristics. Yang et al. (2010a) describes a complete implementation in a prototype tool running on full text. Here we present our experimental results, to illustrate and evaluate the extended methodology.

The rest of the paper is structured as follows. Section 2 introduces the methodology for automatic detection of nocuous ambiguity. Sections 3 and 4 provide details on how the model is applied to coordination and anaphora ambiguity. Experimental setup and results are reported in Section 5, and discussed in Section 6. Section 7 reports on related work. Conclusions and future work are found in Section 8.



*Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1218–1226,
Beijing, August 2010

## 2 Methodology for Nocuous Ambiguity Identification

This section describes the main ideas underpinning our model of ambiguity. We distinguish between structural and interpretative aspects. The former captures the fact that text may have structure (i.e. syntax) which, in principle, permits multiple readings. These are relatively straightforward to identify from the linguistic constructs present in the text. The latter acknowledges that if text is interpreted in the same way by different readers, it has a low risk of being misunderstood. Modelling interpretive aspects requires access to human judgments about texts. Our approach has three elements, which we describe in turn: collection of human judgments; heuristics that model those judgments, and a machine learning component to train the heuristics.

**Human judgments.** We define an ambiguity as nocuous if it gives rise to diverging interpretations. Wasow et al. (2003) suggests that ambiguity is always a product of the meaning that people assign to language, and thus a subjective phenomenon. We capture individual interpretations of instances of ambiguity by surveying participants, asking them for their interpretation. We use this information to decide whether, given some ambiguity threshold, a particular instance is seen as innocuous or nocuous depending on the degree of dissent between judges.

A key concept in determining when ambiguity is nocuous is the *ambiguity threshold*. Different application areas may need to be more or less tolerant of ambiguity (Poesio and Artstein, 2008). For instance, requirements documents describing safety critical systems should seek to avoid misunderstandings between stakeholders. Other cases, such as cookbooks, could be less sensitive. Willis et al. (2008)'s general concept of ambiguity threshold sought to implement a flexible tolerance level to nocuous ambiguity. Given an instance of ambiguous text, and a set of judgments as to the correct interpretation, the *certainty* of an interpretation is the percentage of readers who assign that interpretation to the text. For example, in Table 1 below (sec. 3.1), the certainty of the two interpretations, HA and LA of expression (a) are 12/17=71% and 1/17=5.9% respectively. Here, an expression shows *nocuous ambiguity* if none of the possible interpretations have a certainty exceeding the chosen threshold. Later in this section, we will describe further experiments with alternative, finer grained approaches to setting and measuring thresholds, that affect the classifier's behaviour.

**Heuristics.** Heuristics capture factors that may favour specific interpretations. Each heuristic embodies a hypothesis, drawn from the literature, about a linguistic phenomenon signifying a preferred reading. Some use statistical information (e.g., word distribution information obtained from a generic corpus, the BNC[1], using the Sketch Engine[2]). Others flag the presence of surface features in the text, or draw on semantic or world knowledge extracted from linguistic resources like WordNet[3] or VerbNet[4].

**Machine learning (ML).** Individual heuristics have limited predictive power: their effectiveness lies in their ability to operate in concert. Importantly, the information they encapsulate may be interdependent. We harness this by using ML techniques to combine the outputs of individual heuristics. ML is an established method for recognizing complex patterns automatically, making intelligent decisions based on empirical data, and learning of complex and nonlinear relations between data points. Our model uses supervised learning ML techniques, deducing a function from training data, to classify instances of ambiguity into nocuous or innocuous cases. The classifier training data consists of pairs of input objects (i.e. vectors made up of heuristics scores) and desired outputs (i.e. the class labels determined by the distribution of human judgments as captured by thresholds). To select an appropriate ML algorithm for the nocuity classifier, we tested our datasets (described in later sections) on several algorithms in the WEKA[5] package (e.g., decision tree, J48, Naive Bayes, SVM, Logistic Regression, LogitBoost, etc.)

To train, and validate, a nocuity classifier for a particular form of ambiguity, we build a dataset of judgments, and select heuristics that model

[1] http://www.natcorp.ox.ac.uk/
[2] http://sketchengine.co.uk/
[3] http://wordnet.princeton.edu/
[4] http://verbs.colorado.edu/~mpalmer/projects/verbnet.html
[5] http://www.cs.waikato.ac.nz/~ml/index.html

the information underlying the human judgements about a preferred interpretation.

We validated the approach on two forms of ambiguity. Sections 3 and 4 discuss how the methodology is applied to forms of coordination and anaphoric ambiguity, and evaluate the performance of the final classifiers.

## 3 Automatic Identification of Nocuous Coordination Ambiguity

Our previous work on nocuous ambiguity has focused on coordination ambiguity: a common kind of structural ambiguity. A coordination structure connects two words, phrases, or clauses together via a coordination conjunction (e.g., '*and*', '*or*', etc) as in the following examples:

(1) *They support a typing system for architectural components and connectors.*

(2) *It might be rejected or flagged for further processing.*

In (1), the coordination construction '*architectural components and connectors*' consists of a **near conjunct** (NC) (i.e. '*components*'), a **far conjunct** (FC) (i.e. '*connectors*'), and the attached **modifier** (M) (i.e. '*architectural*'). This construction allows two bracketings corresponding to high modifier attachment (*[architectural [components and connectors]]*) or low modifier attachement (*[[architectural components] and connector]*). Our aim is to refine Chantree et al (2006) and Willis et al (2008), hence our focus is on the two phenomena they treated: modification in noun phrase coordination (as in (1)) and in verb phrase coordination (as in (2)).

We implemented the heuristics described in the earlier work, and introduced two further ones (local document collocation frequency, and semantic similarity). We used the Chantree et al (2006) dataset of human judgments, but employed the LogitBoost algorithm for implementing the nocuity classifier (rather than the Logistic Regression equation). The following subsections give more detail.

### 3.1 Building a dataset

**Coordination instances.** Our dataset was collected and described by Chantree et al. (2006). It contains 138 coordination instances gathered from a set of requirement documents. Noun compound conjunctions account for the majority (85.5%) of cases (118 instances). Nearly half of these arose as a result of noun modifiers, while there are 36 cases with adjective and 18 with preposition modifiers.

**Human judgment collection.** The coordination instances containing potential ambiguity were presented to a group of 17 computing professionals including academic staff or research students. For each instance, the judges were asked to select one of three options: high modifier attachment (HA), low modifier attachment (LA), or ambiguous (A). Table 1 shows the judgment count for two sample instances. In instance (a) in table 1, the *certainty* of HA is 12/17=71%, and the *certainty* of LA is 1/17=6%. Instance (b) was judged mainly to be ambiguous.

| | Judgments | | |
|---|---|---|---|
| | HA | LA | A |
| (a) *security* and *privacy* requirements | 12 | 1 | 4 |
| (b) electrical *characteristics* and *interface* | 4 | 4 | 9 |

Table 1. Judgment count for the sample instances (HA=high attachment; LA=low attachment; and A=Ambiguous)

We set an ambiguity threshold, $\tau$, to determine whether the distribution of interpretations is nocuous or innocuous with respect to that particular $\tau$. If the *certainty* of neither interpretation, HA or LA, exceeds the threshold $\tau$, we say this is an instance of *nocuous* coordination. Otherwise it is *innocuous*. Here, (a) displays nocuous ambiguity for $\tau$>71%.

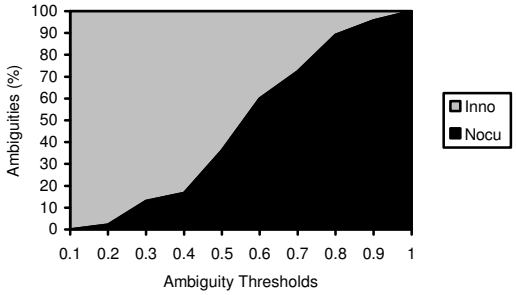


Figure 1. Proportions of interpretations at different ambiguity thresholds in the coordination instances

Figure 1 shows the systematic relationship between ambiguity threshold and the incidence of nocuous ambiguity in the dataset. Low thresholds can be satisfied with a very low certainty scores resulting in few instances being considered nocuous. At high thresholds, almost all instances are classified as nocuous unless the judges report a consensus interpretation.

### 3.2 Heuristics to predict Nocuity

Each heuristic tests a factor favouring a high or low modifier attachment (HA or LA). We implemented and extended Willis et al. (2008).

**Coordination matching** favours HA when the head words of near and far conjuncts are frequently found coordinated in a general corpus like BNC, suggesting they may form a single syntactic unit.

**Distribution similarity** measures how often two words are found in the same contexts. It favours HA where it detects a strong distributional similarity between the headwords of the two conjuncts, suggesting these form a syntactic unit (Kilgariff 2003).

**Collocation frequency** favours LA when the modifier is collocated much more frequently with the headword of the near conjunct than the far conjunct, in the document, or in the BNC.

**Morphology** favours HA when the conjunct headwords share a morphological marker (suffix) (Okumura and Muraki 1994).

**Semantic similarity** favours HA when the conjunct headwords display strong similarity in the taxonomic structure in WordNet[6].

### 3.3 Nocuity classification

To train, and test, the nocuity classifier, each ambiguity training/test instance is represented as an attribute-value vector, with the values set to the score of a particular heuristic. The class label of each instance (nocuous (Y) or innocuous (N) at a given ambiguity threshold) is determined by the *certainty* measure as discussed earlier. We selected the LogitBoost algorithm for building the classifier, because it outperformed other candidates on our training data than. To determine whether a test instance displays nocuity or not, we presented its feature vector to the classifier, and obtained a predicted class label (Y or N).

## 4 Automatic Identification of Nocuous Anaphora Ambiguity

An **anaphor** is an expression referring to an **antecedent**, usually a noun phrase (NP) found in the preceding text. A*naphora ambiguity* occurs when there are two or more candidate antecedents, as in example (3).

(3) *The procedure shall convert the 24 bit image to an 8 bit image, then display **it** in a dynamic window.*

In this case, both of the NPs, '*the 24 bit image*' and '*an 8 bit image*', are considered potential candidate antecedents of the anaphor '*it*'.

Anaphora ambiguity is difficult to handle due to contextual effects spread over several sentences. Our goal is to determine whether a case of anaphora ambiguity is nocuous or innocuous, automatically, by using our methodology.

### 4.1 The building of the Dataset

**Anaphora instances.** We collected 200 anaphora instances from requirements documents from RE@UTS website[7]. We are specifically concerned with 3rd person pronouns, which are widespread in requirements texts. The dataset contains different pronoun types. Nearly half the cases (48%) involve subject pronouns, although pronouns also occurred in objective and possessive positions (15% and 33%, respectively). Pronouns in prepositional phrases (e.g., '*under it')* are rarer (4% - only 8 instances).

**Human judgment collection.** The instances were presented to a group of 38 computing professionals (academic staff, research students, software developers). For each instance, the judges were asked to select the antecedent from the list of NP candidates. Each instance was judged by at least 13 people. Table 2 shows an example of judgment counts, where 12 out of 13 judges committed to '*supervisors*' as the antecedent of *'they'*, whereas 1 chose '*tasks*'.

| 1. Supervisors may only modify tasks **they** supervise to the agents they supervise. | | |
|---|---|---|
| | Response Percent | Response Count |
| (a) supervisors | 92.3% | 12 |
| (b) tasks | 7.7% | 1 |

Table 2. Judgment count for an anaphora ambiguity instance.

**Ambiguity threshold**. Given an anaphor, the interpretation *certainty* of a particular NP candidate is calculated as the percentage of the judgments for this NP against the total judgments for the instance. For example, consider the example in Table 2. The certainty of the NP '*supervisors*'

[6] Implemented by the NLP tool - Java WordNet Similarity Library. http://nlp.shef.ac.uk/result/software.html

[7] http://research.it.uts.edu.au/re/

is 12/13=92.3% and the certainty of the NP '*tasks*' is 1/13=7.7%. Thus, at an ambiguity threshold of, for instance, $\tau = 0.8$, the ambiguity in Table 2 is *innocuous* because the agreement between the judges exceeds the threshold.

Figure 2 shows the relationship between ambiguity threshold and occurrence of nocuous ambiguity. As in Figure 1, the number of nocuous ambiguities increases with threshold τ. For high thresholds (e.g., $\tau \geq 0.9$), more than 60% of instances are classified as nocuous. Below threshold ($\tau \leq 0.4$), fewer than 8 cases are judged nocuous. Also, comparing Figures 1 and 2 would appear to suggest that, in technical documents, anaphora ambiguity is less likely to lead to misunderstandings than coordination.

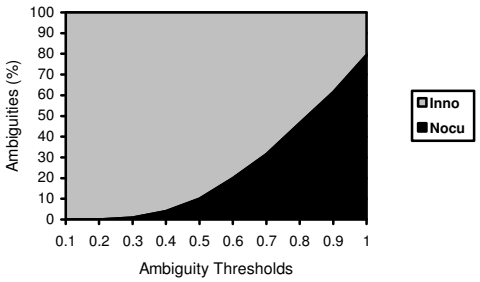


Figure 2. Proportions of interpretations at different ambiguity thresholds in the anaphora instances.

## 4.2 Antecedent Preference Heuristics

Drawing on the literature on anaphoric reference, we developed 12 heuristics of three types: related to *linguistic properties* of text components, to *context and discourse* information, or to *statistical* information drawn from standard corpora. Yang et al. (2010b) gives more detail. A heuristic marks candidate antecedents which it favours, or disfavours. For instance, heuristics favour definite NPs as antecedents, candidate NPs which agree in number and syntactic role with the anaphor, and those which share a syntactic collocation pattern in the text. They also favour those which respect the semantic constraints (e.g., animacy) propagated from subcategorisation information, and reward proximity to the anaphor. They disfavour candidate antecedents that occur in prepositional phrases, and those occupying a syntactic role distinct from the anaphor. Note: not all NPs are marked by all heuristics, and some heuristics are interdependent.

## 4.3 Nocuous Ambiguity Identification

Unlike coordination ambiguity, where judges chose for high or low modifier attachment, anaphora have scope over a variable set of potential antecedents, depending on each particular instance. To accommodate this, we developed an *antecedent* classifier which assigns a weighted antecedent tag to each NP candidate associated with an instance. Tag information is used subsequently to predict the whether the instance displays nocuous ambiguity.

The antecedent classifier is built using the Naive Bayes algorithm within the WEKA package and is trained to return three classes of candidate antecedent: *positive (Y)*, *questionable (Q)*, or *negative (N)*. In an *innocuous* case, a candidate NP will be classed as *Y* if its interpretation certainty exceeds the threshold set by τ, and tagged as *N* otherwise; in a *nocuous* case, it will be classed as *N* if its certainty is 0%, and classified as *Q* otherwise.

| 1. The LPS operational scenarios represent sequences of activities performed by operations personnel as **they** relate to the LPS software. | | |
|---|---|---|
| | Response | Label |
| (a) the LPS operational scenarios | 33.3% | Q |
| (b) sequences of activities | 66.7% | Q |
| (c) activities | 0% | N |
| (d) operations personnel | 0% | N |

Table 3. The determination of antecedent label for the NP candidates in a NOCUOUS ambiguity case ($\tau = 0.8$)

| 2. Testing performed to demonstrate to the acquirer that a CSCI system meets **its** specified requirements. | | |
|---|---|---|
| | Response Percent | Class Label |
| (a) Testing | 0% | N |
| (b) the acquirer | 16.7% | N |
| (c) a CSCI system | 83.3% | Y |

Table 4. The determination of antecedent label for the NP candidates in a INNOCUOUS ambiguity case ($\tau = 0.8$)

| | Antecedent Class Label | | |
|---|---|---|---|
| | Y | Q | N |
| $\tau = 0.5$ | 181 | 54 | 623 |
| $\tau = 0.6$ | 160 | 99 | 599 |
| $\tau = 0.7$ | 137 | 149 | 572 |
| $\tau = 0.8$ | 107 | 209 | 542 |
| $\tau = 0.9$ | 77 | 261 | 520 |
| $\tau = 1.0$ | 41 | 314 | 503 |

Table 5. The distribution of three antecedent class label at different ambiguity thresholds

Table 3 and 4 illustrate antecedent labels for NP antecedent candidates in a nocuous and innocuous case. Candidates (a) and (b) in Table 3 are labeled Q because their certainty falls below the threshold ($\tau = 0.8$). For the same threshold, candidate (c) in Table 4 is tagged as *Y*. Table 5

shows the distribution of tags at certainty thresholds $\tau \geq 0.5$ for all (858) candidate antecedents in our sample.

Our intended application is a system to alert experts to risk of misunderstandings. This suggests we should emphasise recall even at the expense of some precision (Berry et al. 2003). We developed two versions of the algorithm that determines whether an instance is nocuous or not, depending on the contribution made by its antecedent candidates tagged *Y*. We relax constraints by introducing two concepts: a weak positive threshold $W_Y$ and a weak negative threshold $W_N$ set at 0.5 and 0.4, respectively[8]. The rationale for weak thresholds is that antecedent preference reflects a spectrum with *Y* (high), *Q* (medium), and *N* (low). Weak positive and negative thresholds act as buffers to the *Q* area. Antecedent NPs that fall in the $W_Y$ or $W_N$ buffer area are treated as possible false negative (FN) for the classification of the label *Q*. An antecedent tag *Y/N* is labeled as weak positive or negative depending on these thresholds. The algorithm for identifying nocuous ambiguity is given in Figure 3. It treats as innocuous those cases where the antecedent label list contains one clear *Y* candidate, whose certainty exceeds all others by a margin.

Given an anaphora ambiguity instance with multiple potential NPs, the antecedent classifier returns a label list, $R = \{r_1, r_2, \ldots, r_n\}$, for individual NPs.

**Parameters:**
1) $W_Y$ - the threshold for the *weak positive* label. The label *Y* is viewed as *weak positive* when the positive prediction score $r_i < W_Y$
2) $W_N$ - the threshold for the *weak negative* label. The label *N* is viewed as *weak negative* when the negative prediction score $r_i < W_N$

**Procedure:**
**if** the label list *R* contains
(one *Y*, no *Q, one or more N* )
**or**
(no *Y*, one *Q*, one or more *N* but not *weak negative* )
**or**
(one *Y* but not *weak positive*, any number of *Q or N*)
**then**
the ambiguity is INNOCUOUS
**else**
the ambiguity is NOCUOUS

Figure 3. The algorithm for nocuous ambiguity identification

## 5 Experiments and Results

In all experiments, the performance was evaluated using 5-fold cross-validation, using standard measures of Precision (P), Recall (R), F-measure (F), and Accuracy. We use two naive baselines: BL-1 assumes that all ambiguity instances are *innocuous*; BL-2 assumes that they are all *nocuous*. For fair comparison against the baselines, for both forms of ambiguity, we only report the performance of our ML-based models when the incidence of nocuous ambiguities falls between 10% ~ 90% of the set (see Figures 1 and 2). We first report our findings for the identification of nocuous coordination ambiguities and then discuss the effectiveness of our model in distinguishing possible nocuous ambiguities from a set of ambiguity instances.

### 5.1 Nocuous Coordination Ambiguity Identification

Willis et al (2008) demonstrated the ability of their approach to adapt to different thresholds by plotting results against the two naïve base lines. Since we extended and refined their approach described we plot our experimental results (CM-1), for comparison, using the same measures, against their evaluation data (CM-2), in Figure 4.

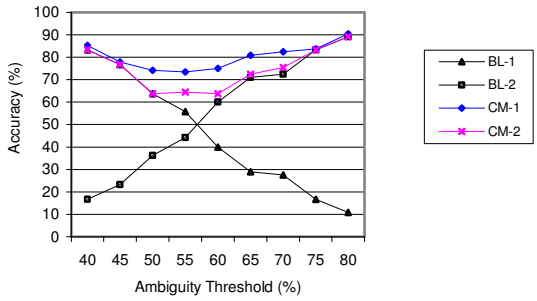


Figure 4. The performance comparison of the ML-based models, CM-1 and CM-2, to the two baseline models, BL-1 and BL-2, in nocuous coordination ambiguity identification.

Our CM-1 model performed well with an accuracy of above 75% on average at all ambiguity threshold levels. As expected, at very high and very low thresholds, we did not improve on the naive baselines (which have perfect recall and hence high accuracy). The CM-1 model displayed its advantage when the ambiguity threshold fell in the range between 0.45 and 0.75 (a significantly wider range than reported for CM-2 Willis et al (2008)). CM-1 maximum improvement was achieved around the 58% crossover point where the two naïve baselines intersect and our model achieved around 21% increased accu-

[8] Weak positive and negative thresholds are set experimentally.

racy. This suggests that the combined heuristics do have strong capability of distinguishing nocuous from innocuous ambiguity at the weakest region of the baseline models.

Figure 4 also shows that, the CM-1 model benefitted from the extended heuristics and the LogitBoost algorithm with an increased accuracy of around 5.54% on average compared with CM-2. This suggests that local context information and semantic relationships between coordinating conjuncts provide useful clues for the identification of nocuous ambiguity. Furthermore, the LogitBoost algorithm is more suitable for dealing with a numeric-attribute feature vector than the previous Logistic Regression algorithm.

### 5.2 Nocuous Anaphora Ambiguity Identification

We report on two implementations: one with weak thresholds (AM-1) and one without (AM-2). We compare both approaches using the baselines, BL-1 and BL-2 (in Figure 5). It shows that AM-1 and AM-2 achieve consistent improvements on baseline accuracy at high thresholds ($\tau \geq 0.75$). Here also, the improvement maximises around the 83% threshold point where the two baselines intersect. However, the ML-based models perform worse than BL-1 at the lower thresholds ($0.5 \leq \tau \leq 0.7$). One possible explanation is that, at low thresholds, performance is affected by lack of data for training of the *Q* class label, an important indicator for nocuous ambiguity (see Table 5). This is also consistent with the ML models performing well at higher thresholds, when enough nocuous instances are available for training.

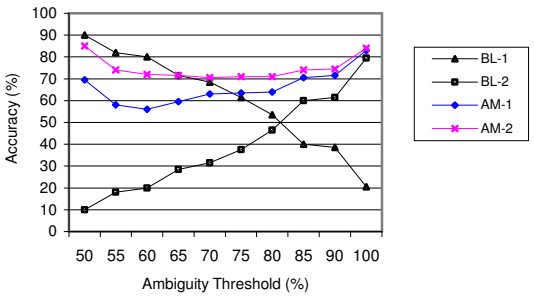


Figure 5. The performance comparison of the ML-based models, AM-1 and AM-2, to the two baseline models, BL-1 and BL-2, in nocuous anaphora ambiguity identification.

Figure 5 further shows that the model with weak thresholds (AM-1) did not perform as well as the model without weak thresholds (AM-2) on accuracy. Although both models perform much better than the baselines on precision (more experimental results are reported in Yang et al. (2010b)), the actual precisions for both models are relatively low, ranging from 0.3 ~ 0.6 at different thresholds. When the AM-1 model attempts to discover more nocuous instances using weak thresholds, it also introduces more false positives (innocuous instances incorrectly classed as nocuous). The side-effect of introducing false positives for AM-1 is to lower accuracy. However, the AM-1 model outperforms both AM-2 and BL-2 models on F-measure (Figure 6), with an average increase of 5.2 and 3.4 percentage points respectively. This reveals that relaxing sensitivity to the ambiguity threshold helps catch more instances of nocuous anaphora ambiguity.

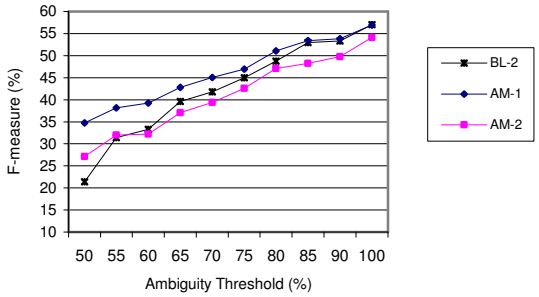


Figure 6. The performance comparison of the ML-based models, AM-1 and AM-2, to the baseline model BL-2 (naïve nocuous)

## 6 Discussions

We presented judges with sentences containing ambiguities without any surrounding context, even though contextual information (e.g., discourse focus) clearly contributes to interpretation. This is a weakness in our data collection technique. Besides contextual information, van Deemter's Principle of Idiosyncratic Interpretation (1998) suggests that some factors, including the reader's degree of language competence, can affect perceptions of ambiguity. Similarly, familiarity with a domain, including tacit specialist information (Polanyi, 1966), and the extent to which this is shared by a group, will have an effect on the extent to which stakeholders arrive at diverging interpretations.

In our case, we extracted instances from requirements documents covering several techni-

cal domains. Judgements are sensitive to the backgrounds of the participants, and the extent to which stakeholder groups share such a background. Also, we used several large, generic NL resources, including the BNC and WordNet. The performance of several heuristics would change if they drew on domain specific resources. Different interpretations may be compatible, and so not necessarily contribute to misunderstanding.

Finally, we used different machine learning algorithms to tackle different types of ambiguity instances: LogitBoost for coordination ambiguity and Naive Bayes for anaphora ambiguity. The main reason is that coordination heuristics returned numeric values, whereas the anaphora heuristics were Boolean. Our method assumes tailoring of the ML algorithm to the choice of heuristic. These limitations indicate that the methodology has a high degree of flexibility, but also that it has several interdependent components and background assumptions that have to be managed if an application is to be developed.

## 7 Related Work

Many researchers have remarked on the fact that some ambiguities are more likely than others to lead to misunderstandings, and suggested classifying them accordingly. Poesio (1996) discussed cases where multiple readings are intended to coexist, and distinguished between language inherent and human disambiguation factors from a philosophical perspective. His notion of '*perceived ambiguity*' suggests that human perceptions are what actually cause an ambiguity to be misunderstood. Van Deemter's (2004) '*vicious ambiguity*' refers to an ambiguity that has no single, strongly preferred interpretation. He proposed quantifying 'viciousness' using probabilities taken from corpus data. Van Rooy (2004) defined a notion of '*true ambiguity*': a sentence is truly ambiguous only if there are at least two interpretations that are optimally relevant. These last two approaches rely on probability analysis of language usage, and not directly on human perception, which we believe to be the key to evaluating ambiguity. Our work differs in that it takes into account the distribution of interpretations arrived at by a group of human judges engaged with a text. Our model treats ambiguity not as a property of a linguistic construct or a text, or a relation between a text and the perceptions of a single reader, but seeks to understand the mechanisms that lead to misunderstandings between people in a group or process.

Poesio *et al* (2006) have pointed out that disambiguation is not always necessary; for instance, in some complex anaphora cases, the final interpretation may not be fully specified, but only 'good enough'. Our work does not attempt disambiguation. It seeks to highlight the risk of multiple interpretations (whatever those are).

## 8 Conclusions and Future Work

We have presented a general methodology for automatically identifying nocuous ambiguity (i.e. cases of ambiguity where there is a risk that people will hold different interpretations) relative to some tolerance level set for such a risk. The methodology has been implemented in a ML based architecture, which combines a number of heuristics each highlighting factors which may affect how humans interpret ambiguous constructs. We have validated the methodology by identifying instances of nocuous ambiguity in coordination and anaphoric constructs. Human judgments were collected in a dataset used for training the ML algorithm and evaluation. Results are encouraging, showing an improvement of approximately 21% on accuracy for coordination ambiguity and about 3.4% on F-measure for anaphora ambiguity compared with naive baselines at different ambiguity threshold levels. We showed, by comparison with results reported in Willis et al (2008) that the methodology can be fine tuned, and extended to other ambiguity types, by including different heuristics.

Our method can highlight the risk of different interpretations arising: this is not a task a single human could perform, as readers typically have access only to their own interpretation and are not routinely aware that others hold a different one. Nonetheless, our approach has limitations, particularly around data collection, and for anaphora ambiguity at low thresholds. We envisage further work on the implementation of ambiguity tolerance thresholds

Several interesting issues remain to be investigated to improve our system's performance and validate its use in practice. We need to explore how to include different and complex ambiguity types (e.g., PP attachment and quantifier scop-

ing), and investigate whether these are equally amenable to a heuristics based approach.

## Acknowledgement

This work is supported financially by UK EPSRC for the MaTREx project (EP/F068859/1), and Irish SFI for the grant 03/CE2/I303_1.

## References

Daniel M. Berry, Erik Kamsties, and Michael M. Krieger. 2003. From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity. Technical Report, School of Computer Science, University of Waterloo.

Stephen Boyd, Didar Zowghi, and Alia Farroukh. 2005. Measuring the Expressiveness of a Constrained Natural Language: An Empirical Study. In *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05)*, Washington, DC, pages 339-52.

Eric Brill and Philip Resnik. 1994. A Rule-Based Approach to Prepositional Phrase Attachment Disambiguation. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 1198-204.

Francis Chantree, Bashar Nuseibeh, Anne de Roeck, and Alistair Willis. 2006. Identifying Nocuous Ambiguities in Natural Language Requirements. In *Proceedings of 14th IEEE International Requirements Engineering Conference (RE'06)*, Minneapolis, USA, pages 59-68.

Adam Kilgarriff. 2003. Thesauruses for Natural Language Processing. In *Proceedings of NLP-KE*, pages 5-13.

Preslav Nakov and Marti Hearst. 2005. Using the Web as an Implicit Training Set: Application to Structural Ambiguity Resolution. In *Proceedings of HLT-NAACL'05*, pages 835-42.

Akitoshi Okumura and Kazunori Muraki. 1994. Symmetric Pattern Matching Analysis for English Coordinate Structures. In *Proceedings of the 4th Conference on Applied Natural Language Processing*, pages 41-46.

Massimo Poesio. 1996. Semantic Ambiguity and Perceived Ambiguity In *Semantic Ambiguity and Underspecification* edited by K. van Deemter and S. Peters, pages 159-201.

Massimo Poesio and Ron Artstein. 2008. Introduction to the Special Issue on Ambiguity and Semantic Judgements. *Research on Language & Computation* 6: 241-45.

Massimo Poesio, Patick Sturt, Ron Artstein, and Ruth Filik. 2006. Underspecification and Anaphora: Theoretical Issues and Preliminary Evidence. *Discourse Processes* 42(2): 157-75.

Michael Polanyi. 1966. *The Tacit Dimension*. RKP, London.

Kees van Deemter. 1998. Ambiguity and Idiosyncratic Interpretation. *Journal of Semantics* 15(1): 5-36.

Kees van Deemter. 2004. Towards a Probabilistic Version of Bidirectional Ot Syntax and Semantics. *Journal of Semantics* 21(3): 251-80.

Robert van Rooy. 2004. Relevance and Bidirectional Ot. In *Optimality Theory and Pragmatic*, edited by R. Blutner and H. Zeevat, pages 173-210.

Thomas Wasow, Amy Perfors, and David Beaver. 2003. The Puzzle of Ambiguity. In *Morphology and the Web of Grammar: Essays in Menory of Steven G. Lapointe*, edited by O. Orgun and P. Sells.

Alistair Willis, Francis Chantree, and Anne De Roeck. 2008. Automatic Identification of Nocuous Ambiguity. *Research on Language & Computation* 6(3-4): 1-23.

Hui Yang, Alistair Willis, Anne de Roeck, and Bashar Nuseibeh. 2010a. Automatic Detection of Nocuous Coordination Ambiguities in Natural Language Requirements. *In Proceedings of the 25th IEEE/ACM International Conference on Automated Software Engineering Conference (ASE'10)*. (In press)

Hui Yang, Anne de Roeck, Alistair Willis, and Bashar Nuseibeh. 2010b. Extending Nocuous Ambiguity Analysis for Anaphora in Natural Language Requirements. *In Proceedings of the 18th International Requirements Engineering Conference (RE'10)*. (In press)