2010 18th IEEE International Requirements Engineering Conference

Extending Nocuous Ambiguity Analysis for Anaphora in Natural Language Requirements

Hui Yang[1] Anne de Roeck[1] Vincenzo Gervasi[2] Alistair Willis[1] Bashar Nuseibeh[1, 3]

[1] Department of Computing, The Open University, UK
[2] Department of Computer Science, University of Pisa, Italy
[3] Lero, University of Limerick, Ireland
e-mail: {h.yang, a.g.willis, a.deroeck, b.nuseibeh}@open.ac.uk; gervasi@di.unipi.it

Abstract. This paper presents an approach to automatically identify potentially nocuous ambiguities, which occur when text is interpreted differently by different readers of requirements written in natural language. We extract a set of anaphora ambiguities from a range of requirements documents, and collect multiple human judgments on their interpretations. The judgment distribution is used to determine if an ambiguity is nocuous or innocuous. We investigate a number of antecedent preference heuristics that we use to explore aspects of anaphora which may lead a reader to favour a particular interpretation. Using machine learning techniques, we build an automated tool to predict the antecedent preference of noun phrase candidates, which in turn is used to identify nocuous ambiguity. We report on a series of experiments that we conducted to evaluate the performance of our automated system. The results show that the system achieves high recall with a consistent improvement on baseline precision subject to some ambiguity tolerance levels, allowing us to explore and highlight realistic and potentially problematic ambiguities in actual requirements documents.

Keywords: nocuous ambiguity; NL requirements; anaphora ambiguity; antecedent preference heuristics; machine learning

I. INTRODUCTION

The vast majority of requirements are in practice written in natural language (NL) [3], and thus suffer from typical NL problems such as ambiguity. Ambiguity is a phenomenon inherent in natural language, and occurs when a linguistic expression can be understood in two or more possible ways by different readers. Ambiguity has often been considered a potentially dangerous attribute of requirements [4], since these requirements are then used to develop specifications that convey univocally the desired behaviour of a system to be developed. So, for example, if a system is developed and tested according to an interpretation different from that of the customer, it might not be accepted after customer validation; or if a user manual is written based on an interpretation of the requirements different from that of the developers, it can document erroneously how the system works.

Still, ambiguity is a fundamental feature of human languages, and it exists for good reasons – including, the need to convey multiple acceptable meanings, or to leave leeway for negotiations, or for economy of expression when it is assumed that common sense or domain knowledge will lead every stakeholder to select the same (correct) interpretation. Previous work on ambiguity in RE has attempted to address the problem from at least two perspectives:

- Providing users with a restricted NL [12] or handbook [2, 3] to assist with writing less ambiguously;
- Detecting ambiguity by examining the text using lexical, syntactic, or semantic information, or with the help of quality metrics [13, 14, 16, 17].

Our approach to ambiguity analysis in NL requirements differs from earlier work. Our research aims at helping requirements analysts to focus on identifying ambiguity instances in requirements that are *likely* to lead to misunderstandings between stakeholders, while discounting those that are unlikely to cause misunderstandings. The analyst can then perform a more careful analysis of risk and impact of those problematic cases (an analysis that is left to human judgement), and to decide whether more elicitation is needed, or other risk-control measures need to be taken[1]. We do not consider ambiguity just as a property of a text, but add a categorization of *nocuous* and *innocuous* ambiguity based on the interpretations held by a group of readers of that text [6, 22]. We observe that not all cases of structural ambiguity are actually harmful (in fact, the majority are not) because readers will interpret them in the same way, and so these are not worthy of the attention of the analyst. We call these *innocuous* because the risk of misunderstanding between all the relevant readers (i.e. those involved in the development process) is low, and as *nocuous* those cases in which the structural ambiguity really manifest itself through different, conflicting interpretations by the readers.

An overall objective of our research is to discover the factors that contribute to the preference for particular readings, and to investigate whether the distinction between nocuous and innocuous ambiguity is evident, and can be implemented in a computational model. Our approach is to develop a system that can automatically classify ambiguities in NL requirements as nocuous or innocuous, and inform the analyst of the potentially dangerous cases, to prompt further elicitation. Previously, we proposed a general methodology [24] for automatic identification of nocuous ambiguity,

[1] As with programming, requirements are inherently an ill-conditioned problem; small changes in their interpretation can lead to hugely differently behaviour of the developed systems. We do not address here the issue of estimating the *effect* of different interpretations caused by ambiguity.

1090-705X/10 $26.00 © 2010 IEEE
DOI 10.1109/RE.2010.14

which we use here to guide our research. In previous work [6, 22], we also focused on coordination ambiguity, and developed a set of heuristics to automatically predict whether a coordination ambiguity may be misinterpreted given an ambiguity threshold. We also implemented a prototype tool for automatic identification of nocuous coordination ambiguity [25]. In this paper, we investigate a further, prominent ambiguity type in NL requirements, namely *anaphora ambiguity* (exemplified by extensive use of pronouns – such as 'it', 'they' – in the many requirements documents we studied). Anaphora ambiguity is considered hard to resolve because contextual dependencies have to be taken into account [3], and because the structures leading to it may be spread over several sentences or statements. Our goal is therefore to build a system that can determine automatically whether an instance of anaphora ambiguity is nocuous or innocuous by using a set of heuristics that contribute evidence to support a particular antecedent candidate, which we subject to a machine learning (ML) based classifier.

To train our system, and to validate its performance, we extracted a set of anaphora ambiguity instances from requirements documents. For each instance, we collected multiple human judgements, whose distribution was used to determine if the ambiguity is either nocuous or innocuous. We also used the notion of *ambiguity threshold* to indicate the degree of nocuous ambiguity that can be tolerated with reference to a general population of requirements analysts for a given domain of application, to account for the fact that some domains are associated with different risk tolerance levels.

The rest of the paper is organized as follows: Section II provides some background on anaphora ambiguity, with relevant anaphora examples. In Section III we give more details about the building of our dataset. Section IV presents three types of antecedent preference heuristics which contribute to predicting the preferred antecedent candidate. In Section V, we introduce the procedure for nocuous ambiguity identification. Our experimental setup and results are reported in Section VI, and Section VII addresses potential threats to validity. Section VIII reflects on related work, and our conclusions and plans for future work are discussed in Section IX.

II. ANAPHORA AMBIGUITY

An *anaphor*[2] is an expression used, in language, to refer to another expression. Personal pronouns, such as *he, she, it, those, our,* etc. are examples of anaphora, or anaphoric expressions. In this paper, we are specifically concerned with the 3rd person pronouns such as 'it', 'its', 'they', 'them', and 'their', because we have found them to be widespread in requirements documents.

The expression an anaphor refers to is called an *antecedent*. Antecedents for personal pronoun anaphora are noun phrases[3] (NPs) found elsewhere in the text, usually preceding the anaphor itself. For instance, consider the example below[4]:

E1: *A prototype exists and **it** will be examined for reuse.*

In (E1), the pronoun '*it*' is an anaphor and *'A prototype'* is the antecedent NP. The interpretation of (E1) states that the prototype will be examined for re-use. The intuition is that the anaphor is a placeholder for the antecedent that has been mentioned elsewhere.

An *anaphoric ambiguity* occurs when the text offers two or more appropriate antecedent candidates either in the same sentence or a preceding one, as in example (E2)

E2. *The procedure shall convert the 24 bit image to an 8 bit image, then display **it** in a dynamic window.*

In this case, both interpretations are legitimate: a procedure that displays the 24 bit image would meet the requirement, as would one that displays the 8 bit image. It is possible for different stakeholders to commit, legitimately, to different, incompatible reading of the same requirement. Now consider examples (E3) and (E4), both of which have more than one candidate antecedent for the anaphor *'it'*.

E3. *The MANUAL schema models the system behavior when **it** is in manual override mode.*

E4. *A prototype exists for this process and **it** will be examined for reuse.*

We asked 13 people with software engineering backgrounds to identify which NP they thought the anaphor referred to in (E3). 7 committed to "*the MANUAL schema*", whereas 6 chose '*the system behavior*'. Hence, we consider the instance (E3) as a typical *nocuous* ambiguity case for study. In contrast, for (E4), all judges agreed that '*a prototype*' is the antecedent. This illustrates the difference between nocuous and innocuous ambiguity. Both (E3) and (E4) contained more than one possible antecedent, but in the case of (E3) the interpretations of stakeholders significantly diverged, whereas all stakeholders committed to the same antecedent in (E4). We identify (E3) as a case of *nocuous anaphoric ambiguity* and (E4) as a case of *innocuous anaphoric ambiguity*.

Three further points. Highlighting the presence of possible structural ambiguities in requirements documents is not an effective approach as they are rife. Also, traditional approaches that try to address ambiguity issues by automatically determining the correct antecedent in (E3) are inappropriate since stakeholders would still hold incompatible interpretations. Finally, since we are trying to avoid different stakeholders committing to incompatible interpretations, no single person has access to what other stakeholders' interpretations are. Our approach is to alert users to where nocuous ambiguities might occur, and augment (rather than attempt to replicate) human capability.

[2] Plural: *anaphora*

[3] For simplicity, phrases containing nouns.

[4] Our examples are adapted (in many cases abbreviated) from our collection of requirements documents. We will render anaphora in bold and underline candidate antecedents.

III. THE BUILDING OF THE DATASET

Anaphora instances in Requirements documents. We collected a set of about 11 requirements documents from RE@UTS web pages[5]. The documents specify systems from a variety of application domains, including transportation, engineering, communication, and web applications. In this dataset, we located a set of 200 anaphora instances containing different types of pronouns. Each instance contains one or two sentences (i.e. the current sentence that the pronoun appears and the preceding one) depending on the position of the pronoun in the sentence, and has two or more possible noun phrase (NP) candidates for the antecedent of an anaphor. Nearly half of the cases (48%) arise as a result of subject pronouns (e.g., '*it*', '*they*') although objective pronouns (e.g., '*them*', '*it*') and possessive pronouns (e.g., '*its*', '*their*') also accounted for a significant number (15% and 33%, respectively). The cases related to prepositional pronouns (e.g., '*under it*') are relatively rare (4% only - 8 instances in the whole dataset).

Human judgment collection. The anaphora instances containing potential ambiguity were split into five separate surveys (i.e. each survey contains 40 anaphora instances), which were conducted on a web site[6] that we designed. The anaphora instances were presented to a group of 38 computing professionals who were academic staff, research students, or software developers. Among these judges, 25 were native English speakers. An introduction was given to explain the task and to illustrate how to make judgments with an anaphora sample before the judges were asked to record their interpretations for the actual ambiguous examples. The NP candidates and the anaphor were highlighted in the examples presented to the judges (See the example in Table I discussed later). For each example, the judges were asked to select one of the NP candidates that they think might fit best to the anaphora reference. To obtain the unbiased opinions from different judges, each judge was allowed to do one survey only once, but was allowed to do different surveys.

Ambiguity threshold. In our dataset, each anaphora ambiguity instance was judged by at least 13 people. In order to predict whether a particular instance displays nocuous ambiguity, we borrow the definition given in Willis, Chantree, and Roeck [22] that introduces the concept of *ambiguity threshold*, and adapt the definition specifically to anaphora ambiguity:

Definition. Given an anaphora instance, *I*, a collection of judgments associated with NP candidates for the antecedent of the anaphor, and an ambiguity threshold τ (where $0.5 < \tau \leq 1.0$):

If there is one NP candidate that has a *certainty* greater than τ, then the instance *I* exhibits *innocuous* ambiguity at the threshold τ. Otherwise, the instance *I* exhibits *nocuous* ambiguity at the threshold τ.

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

[6] http://www.surveymonkey.com/

In this definition, the ambiguity threshold τ is set to be greater than 0.5. The reason for this is because this constraint guarantees that, in an *innocuous* ambiguity case, there is only one NP candidate that can be a potential anaphora reference. Moreover, the *certainty* of a particular NP candidate is calculated as the percentage of the judgments for this NP against the total judgments in the ambiguity instance. For example, consider the example in Table I. The certainty of the NP '*supervisors*' is 11/12=91.7% and the certainty of the NP '*tasks*' is 1/12=8.3%. Thus, at the ambiguity threshold $\tau = 0.8$, the ambiguity in Table I is *innocuous* because there is a clear agreement between the judges on the NP '*supervisors*'.

Figure 1 depicts the relationship between the ambiguity threshold and the exhibition of nocuous ambiguity. It is noted that the number of nocuous ambiguities increases with the ambiguity threshold τ. For high thresholds (e.g., $\tau \geq 0.9$), there are more than 60% of anaphora instances that are classified as nocuous ambiguities. This means that it is not easy for all the judges to agree with the same interpretation in context-dependent cases like anaphora. However, considering the potential errors made by judges through carelessness or by accident, some degree of ambiguity tolerance (e.g., $0.7 \leq \tau \leq 0.9$) should be permitted in such nocuous ambiguity analysis in order to reflect a genuine difference of opinion.

Ambiguities that are nocuous at a lower threshold are more serious than those that are nocuous only at higher thresholds. Thus, τ can be used to establish a ranking among cases of ambiguity, allowing an analyst to focus on the most critical cases first (e.g., when limited time is available for further elicitation).

TABLE I. JUDGMENTS COUNT FOR AN ANAPHORA AMBIGUITY INSTANCE

1. Supervisors may only modify tasks **they** supervise to the agents they supervise.		
	Response Percent	Response Count
(a) supervisors	91.7%	11
(b) tasks	8.3%	1

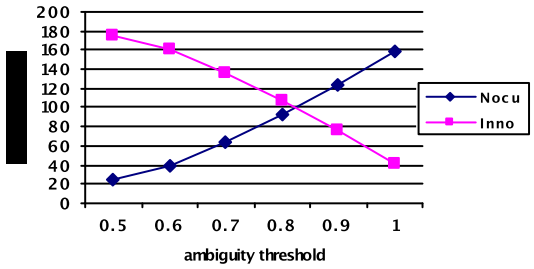

Figure 1. The distributions of nocuous/innocuous ambiguity in the dataset at different ambiguity thresholds τ.

IV. ANTECEDENT PREFERENCE HEURISTICS

When applied to anaphora, our model of nocuous ambiguity involves estimating the risk that an anaphor is interpreted by different stakeholders as referring to different antecedents. This is distinct from disambiguation, which assumes

that there is a single correct interpretation. It is also distinct from identifying when a stakeholder is experiencing a requirement statement as ambiguous: since no stakeholder can be assumed to have access to any other stakeholder's interpretation, the purpose of our model is to highlight ambiguity problems that cannot be identified by a single stakeholder alone.

Our solution is to estimate the risk of divergent interpretations by applying a collection of heuristic rules, each marking a factor which would favor or disfavor an NP as the antecedent of the anaphor. Heuristics run over the text and their output is recorded in a matrix, which is later used by our machine learning algorithm to classify anaphoric ambiguities as nocuous or innocuous. Here, we give a brief description of the heuristics we have developed, with some examples.

Each heuristic implements a preference factor drawn from the literature on anaphoric ambiguity. We have three types: those related to *linguistic properties* of words and sentence components, to *context and discourse* information, and to information on how words and sentence components distribute in the language, as approximated by *statistical information* drawn from corpora, in this case in English. We developed a robust, knowledge-rich approach to predict preferred candidate antecedents of the anaphora. We used lexical, syntactic, and semantic knowledge extracted from several linguistic resources including Stanford Parser[7], WordNet[8], VerbNet[9], and the Sketch Engine[10].

First, we segment our requirements into sentences using a sentence splitter. Then Part-of-Speech (POS) tagging and noun phrase recognition are performed using the Stanford Parser augmented with some simple noun phrase rules, before applying our antecedent preference heuristics. Below we describe the heuristics with some examples.

A. Lingusitic Preference Rules

Our linguistic heuristics record syntactic and semantic clues found useful in identifying antecedents.

Number agreement: In English, anaphors and their antecedents usually must agree in number (i.e. singular or plural). There are exceptions: for instance, certain collective nouns in English (e.g. '*organization*', '*consortium*', '*team*') appear to be singular but can be referred to by a plural anaphor (e.g., '*they')* Similarly, singular antecedents that have been conjoined with *'and'* or *'or'*, as in (E5), take a plural anaphor:

E5. *IT developer and consultant often ask for an exemplary requirements specification as a starting point in **their** specific project.*

Finally, people may use a plural pronoun for a singular antecedent to avoid using a gender-typing pronoun:

E6. *The user will be able to generate a printout to a local printer from **their** browser.*

Although this is not considered best practice usage, it does appear in the requirements documents in our sample and always involves an antecedent that is a person.

In our system, each noun and pronoun is assigned a number property by the Stanford tagger which is checked by the heuristic. We use a syntactic rule to detect the conjunctive NPs. We handle the exceptions by using the *hypernyms* relationships in WordNet, specifically those pointing to collective nouns referring to groups such as 'staff' and 'division', or the individual person like 'employee', 'user'. Moreover, we collected a list of NPs or APs (e.g., 'a lot of' and 'a few', 'many') that modify the plural nouns.

Definiteness: Definite noun phrases are more likely antecedents of anaphors than indefinite ones [19]. An NP is tagged as definite if its head noun is modified by a definite article (e.g., '*the system*'), a demonstrative (e.g., '*this function*'), or a possessive pronoun (e.g., '*its attribute*').

'Non-prepositional' noun phrases: Brennan, Fridedman, and Pollard [5] showed the priority of a syntactic role in antecedent candidate ranking to be 'subject, direct object, indirect object'. In English, subjects and direct objects are not introduced by prepositions (such as *in, to, up*, etc) and our heuristics favour non-prepositional NPs as antecedents. For instance:

E7. *Constraints are conditions about the data that must always be true. **They** are the integrity rules that protect the data in the eventual database*.

Here, the NPs '*constraints*' and '*conditions*' are more likely antecedent candidates because '*the data'* is part of the preposition phrase '*about the data*'.

Syntactic constraint: Empirical evidence suggests that the syntactic roles of antecedent and anaphor often correspond [9]. Our heuristics give preference to candidates that have an identical subject or object role as the anaphor. For example:

E8. *The VCDUs are annotated to reflect the data quality, and any change in the VCID. **They** are also annotated to mark the end of the contact period.*

Here 'They' (the subject) refers to 'The VCDUs' which is also the subject of its sentence.

Syntactic parallelism: Repeating patterns may offer clues about preferred antecedent candidates. For instance, in (E9), both '*This CONOPS*' and '*it*' are the subjects in their own sentence, and are followed by the verb '*describes*', and our heuristics favour *'This CONOPS'* as an antecedent.

E9. *This CONOPS [describes] the mission of the LMI system. **It** also [describes] the functions and characteristics of the system.*

'Non-associated' noun phrases: This heuristic disfavors the NPs that are immediately associated with the anaphor in different syntactic roles. For example:

E10. *The technical interfaces to the EHR system must be documented in such a way that the customers can understand **them** and use **them** for integration*.

[7] http://nlp.stanford.edu/software/lex-parser.shtml
[8] http://wordnet.princeton.edu/
[9] http://verbs.colorado.edu/~mpalmer/projects/verbnet.html
[10] http://sketchengine.co.uk/

In (E10), the NP 'the customers' is unlikely to be the antecedent because it stands in a subject relation to the same verb 'understand' of which 'them' is the object.

Semantic constraint: This heuristic allocates preference status to candidate antecedents that respect semantic constraints. For instance, the verb *'to live'* normally requires an animate agent. In example (E11) our heuristic prefers *'many individuals'* over *'resources and services'*. Semantic properties and constraints, such as animacy, are collected from WordNet and VerbNet.

E11. *Many individuals have difficulty accessing resources and services because* ***they*** *[live] in a geographically remote area.*

Semantic parallelism: This heuristic records a preference for candidate antecedents sharing a semantic role with the anaphor (e.g., subject) as below:

E12. *RDCS [captures] a raw data byte stream after* ***it*** *[receives] the start capture directive from the MACS.*

This heuristic plays a particularly useful role since requirements documents contain many domain-specific terms (e.g., 'RDCS') that are not present in WordNet. On the other hand, verbs are far less likely be highly domain specific, and in the example (E12), both 'capture' and 'receive' are tagged as requiring agents in VerbNet and the heuristic allows us to use this constraint to record the relationship between 'RDCS' and 'it'.

B. Context/Discourse Preferences

The overall performance of antecedent determination can be improved when incorporating features that capture the context/discourse information [5, 19]. These are distinct from the syntactic and semantic clues discussed so far, and concern what is best described as the structure of the *information flow* in a document. We have developed three heuristics.

Centering (discourse focus): This heuristic attempts to capture the most salient entities in a connected chunk of document (e.g., a paragraph or section/subsection) by marking a preference for the most frequently occurring candidate antecedents. We observed that about 86.88% (10037 out of 11552) paragraphs in the dataset contain less than 3 sentences. Here we set the occurrence threshold >=2, where about 31% of the antecedent candidates are chosen.

Sentence Recency: Antecedents and anaphora may occur in different sentences and this heuristic marks, for every candidate antecedent, whether it occurs in the same sentence as the anaphor.

Proximity: Finally, the distance between antecedent and anaphor is significant and this heuristic marks, ranks antecedents for distance to the anaphor. The distance is calculated based on the occurrence position of the antecedent in the left context of the anaphor (see Table II).

C. Statistics/Corpus Preference

Requirements documents tend to be highly specialized and contain terms that are not always part of the general English vocabulary. Comparison with a balanced sample of standard text can contribute useful insight. For instance, example (E11) has two candidate antecedents (*'many individuals'* and *'resources and services'*) for the pronoun *'they'* which is subject to the verb *'live'*. Evidence that one of the two candidates co-occurs frequently with the verb in other contexts increases the possibility that readers will assume it is the antecedent.

Our statistics/corpus preference heuristics mark preferences for candidate antecedents on the basis of co-occurrence patterns observed in the requirements document itself, and in a large corpus – in this case the BNC[11] (British National Corpus). The BNC is a modern corpus containing over 100 million words of English, collated from a variety of sources, including some from the same domains as the documents in our corpus. We use the Sketch Engine to collect collocation frequencies from the BNC.

Finally, we should point out that the above heuristics are antecedent preferences and not obligatory conditions. There might be cases where one or more of the antecedent preferences do not point to the correct antecedent. When all preferences are taken into account, however, the preferred NP candidate is still very likely to be traced down. Moreover, we can easily add more new heuristics or delete some unimportant heuristics if it is needed in order to optimize the system performance.

V. NOCUOUS AMBIGUITY IDENTIFICATION

We used a machine learning algorithm to construct an automated tool - an "antecedent classifier" – that, given a description of an NP candidate, assigns a weighted antecedent tag to the NP candidate. The antecedent tag information in turn is used by our tools to predict whether the anaphora instance displays nocuous ambiguity. Such a tool could be integrated into a requirement authoring environment, similarly to how online spelling/grammar checkers work on present-day word processors, or into a requirements management tool, for more structured analysis. The analyst could then call upon a functionality of the RM tool to identify the set of requirements which include potentially dangerous (i.e. above a desired threshold) cases of ambiguity.

A. Building the Antecedent Classifier

Classification is performed on a discrete-valued feature space. In other words, each training/test instance (i.e. an NP candidate) is represented as an attribute-value vector, where attributes describe properties of the antecedent preferences mentioned earlier (see Table II).

The class labels associated with a training instance, *positive (Y)*, *questionable (Q)*, or *negative (N)*, are obtained based on the response percent of the human judgments collected in the anaphora surveys. The labels in the training set are associated with a particular ambiguity threshold τ.

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

TABLE II. FEATURES FOR THE ANTECEDENT CLASSIFIER. EACH INSTANCE REPRSENTS A SINGLE *NP* CANDIDATE, NP_j, CHARACTERIZED BY 13 FEATURES

Feature Type	Feature	Description
Linguistics	Number agreement	Y if NP_j agree in number; N_P if NP_j does not agree in number but it has a person property; N if NP_j doesn't agree in number; UNKNOWN if the number information cannot be determined.
	Definiteness	Y if NP_j is a definite *NP*; else N.
	Non-prepositional NP	Y if NP_j is a non-prepositional NP; else N.
	Syntactic constraint	Y if NP_j satisfies syntactic constraint; else N.
	Syntactic parallelism	Y if NP_j satisfies syntactic parallelism; else N.
	Non-associated NP	Y if NP_j is a non-associated NP; else N.
	Semantic constraint	Y if NP_j satisfies semantic constraint; else N.
	Semantic parallelism	Y if NP_j satisfies semantic parallelism; else N.
Context	Centering	Y if NP_j occurs in the paragraph more than twice; else N.
	Sentence recency	INTRA_S if NP_j occurs in the same sentence as the anaphor; else INTER_S.
	Proximal	integral value *n*, where *n* means that NP_j is the *n*th *NP* to the anaphor in the right-to-left order
Statistics	Local-based collocation frequency	integral value *n*, where *n* refers to the occurrence number of the matched co-occurrence pattern containing NP_j in local requirements document
	BNC-based collocation frequency	Y if the matched co-occurrence pattern containing NP_j appears in the word list returned by the sketch engine; else N.

Specifically, in an innocuous ambiguity case, only one NP candidate can be marked as *positive* (Y) when the percentage of the judges selecting it as the correct reference is above the ambiguity threshold τ. The remaining NPs are tagged as *negative* (N). On the other hand, in a nocuous ambiguity case, an NP is labeled as *questionable* (Q) only if the percentage selecting it is below τ but greater than 0. Otherwise, it is tagged as *negative* (N).

Tables III and IV illustrate the determination of antecedent labels for the NP candidates in a nocuous/innocuous ambiguity case. For the nocuous ambiguity in Table III, NP candidates (a) and (b) are labeled as the tag 'Q' due to the low response percent under the ambiguity threshold τ, whereas in Table IV, only the NP (c) is tagged with 'Y' because of the high response score, and the others with 'N'.

The antecedent classifier is built using the Naive Bayes algorithm within the WEKA[12] machine learning algorithm package. The antecedent classifier is designed to maximize the accuracy of prediction for antecedent types. The training data is obtained from the dataset of anaphora instances and the associated judgments for each possible antecedent candidate. Table V shows the distribution of three types of anaphoric instances at different agreement certainty thresholds.

B. Applying the Classifier

To determine the antecedent preference of an NP candidate in a test ambiguity case, we create a test instance for the NP, and present the instance to the generated antecedent classifier, which returns a class label of *Positive* (*Y*), *Questionable* (*Q*), or *Negative* (*N*).

C. Using Antecedent Information for Nocuous Ambiguity Identification

As previously discussed, our work emphasizes the discovery of potential nocuous ambiguity in order to report it to a user rather than attempting to disambiguate. The user retains control over the final determination based on some contextual analysis. Berry, Kamsties, and Krieger [2] suggest that it is desirable to maximize the number of potentially nocuous ambiguities found and so one should aim for 100% recall even at the expense of some imprecision. In line with this high recall principle, the algorithm that we developed and deployed (see Figure 2) determines whether an ambiguity is nocuous or innocuous.

In order to obtain more potentially questionable instances of antecedent prediction, we relax the constraints and introduce two concepts, the weak positive threshold W_Y and the weak negative threshold W_N. In our tool, we treat a weak positive or negative instance as a possible questionable instance when its predication score for the class label is below a certain threshold. In our tool, the weak thresholds, W_Y and W_N, are set with 0.5 and 0.4, respectively, which are determined experimentally.

TABLE III. THE DETERMINATION OF ANTECEDENT LABEL FOR THE NP CANDIDATES IN A *NOCUOUS* AMBIGUITY CASE (T=0.8)

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 IV. THE DETERMINATION OF ANTECEDENT LABEL FOR THE NP CANDIDATES IN A *INNOCUOUS* AMBIGUITY CASE (T=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

[12] http://www.cs.waikato.ac.nz/~ml/index.html

TABLE V. THE DISTRIBUTION OF THREE TYPES OF ANAPHORA INSTANCES AT DIFFERENT AMBIGUITY THRESHOLDS

	Antecedent Class Label		
	Y	Q	N
$\tau = 0.6$	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

Algorithm: Nocuous Ambiguity Identification
Given an anaphora ambiguity instance with multiple potential NP candidates, the antecedent classifier returns an antecedent label set, $C \in \{Y, Q, N\}$, with the corresponding predication scores, $R = \{r_1, r_2, \ldots, r_n\}$, for individual NP candidates.

Parameters:
1) W_Y - the threshold for the *weak positive* label. The label *Y* is viewed as *weak positive* when the positive predication 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 2. The algorithm for nocuous ambiguity identification

VI. EXPERIMENTS AND RESULTS

For the purpose of evaluation, we used a standard 5-fold cross validation technique in which, in each iteration, we trained on 80% and tested on 20% of the remaining data. The performance of the task was measured in terms of precision (P), recall (R), and F-measure (F):

$$R = \frac{TP}{TP + FN} \qquad P = \frac{TP}{TP + FP} \qquad F = \frac{(1+\beta^2)PR}{\beta^2 P + R}$$

where *TF* (true positives) is the number of correctly identified *nocuous* ambiguities, FN (false negatives) is the number of nocuous ambiguity not identified by the system, and FP (false positives) is the number of nocuous ambiguities y that are incorrectly identified. Here we use the F2 measure (β=2), which weights recall twice as much as precision in order to emphasize the impact of the recall on performance. All results were averaged across five iterations.

We present our experiments and results as follows: First we conducted a set of experiments to compare the performance of the antecedent classifier at different ambiguity thresholds and discuss this in Section VI.A. We then discuss the impact of various feature types on performance of the antecedent classifier in Section VI.B. In order to evaluate our tool, we compared it to a baseline measurement, and report on our tool performances at different ambiguity thresholds in Section VI.C.

A. Performance of the Antecedent Classifier

The performance of our antecedent classifier trained on the anaphoric instances is reported in Table VI. The best performance of the classifier was achieved at the threshold of 0.6 with an accuracy of 76.24%, whereas the worse performance was obtained at the threshold of 1.0 with a drop of 4.67% in accuracy. A likely reason for the drop in accuracy is the problem of imbalanced distribution among different types of training instances for high thresholds (e.g., $\tau \geq 0.9$), especially for positive instances which are significantly decreased with the increase of the threshold. For example, there are 41 positive instances at the threshold of 1.0 (see Figure 1).

TABLE VI. THE PERFORMANCE OF ANTECEDENT CLASSIFIER AT DIFFERENT THRESHOLDS

	Correct Instance	Incorrect Instance	Accuracy
$\tau = 0.6$	654	204	0.7624
$\tau = 0.7$	637	221	0.7426
$\tau = 0.8$	625	233	0.7283
$\tau = 0.9$	617	241	0.7191
$\tau = 1.0$	614	244	0.7157

B. Impact of Feature Types

We also evaluated the impacts of individual feature types on system performance compared with the full-fledged antecedent classifier with all features. We treated linguistic features as the basic feature type because of the important role that linguistic information plays in anaphora analysis. Table VII shows that the context features or statistical features generally improve the performance of the antecedent classifier when they are incorporated with the linguistic features. Nevertheless, it seems that both types of features appear to have no apparent positive influence on antecedent prediction.

We hypothesized that the best accuracy on all features would be achieved if each type of feature could capture disparate characteristics of anaphora to some extent. The results in Table VIII support this hypothesis: the system performs best when all features are used together with an increase of 2.5% in accuracy.

TABLE VII. THE IMPACT OF FEATURE TYPES ON SYSTEM PERFORMANCE FOR T = 0.8

Feature Type	Correct Instance	Incorrect Instance	Accuracy
Ling	603	255	0.7037
Ling + Ctx	613	236	0.7145
Ling + Stat	612	237	0.7136
All Features	625	233	0.7283

C. System Performance Comparison

We use a random model (baseline) to compare the performance of the proposed ML-based model for nocuous ambiguity identification. In the baseline model, we assume that each recognized ambiguity instance has the potential to be a *nocuous* ambiguity, and is counted as a positive match for the baseline model. Therefore, the baseline model achieves an 'ideal' recall R_{BL} of 100%, and the precision and F-measure are calculated as:

$$P_{BL} = \frac{\#\,Nocuous\ identif\ edby\ Judgments}{total\ \#\ of\ Ambiguities}$$

$$F_{BL} = \frac{(1+\beta^2)P_{BL}R_{BL}}{\beta^2 P_{BL} + R_{BL}} (\beta = 2)$$

The performance comparison among the ML-based model with weak thresholds, the ML-based model without weak thresholds, and the baseline model at different ambiguity thresholds are given in Table VIII. As expected, compared with the baseline model, the ML-based model with weak thresholds performs better with an average increase of 8.06 percentage point on precision and 4.7 percentage points on F2 measure, although the recall is slightly lower compared with the '*ideal*' one of 100%. However, for the ML-based model without the weak thresholds, we see notable improvement in precision, but much larger decreases in recall. Overall, the results suggest that the ML-based model benefits by using antecedent preferences information represented in the antecedent classifier.

Comparing the ML-based model without weak thresholds, we see notable gains in recall, and smaller drops in precision for the ML-based model with weak thresholds. It reveals that the introduction of the weak positive and negative thresholds help catch more instances that fall into the grey areas between positive and questionable instances, or between questionable and negative instances. Those instances are very sensitive to the ambiguity threshold τ.

Furthermore, we also find that the overall performance of the system began to deteriorate as the threshold τ decreases. The possible explanation for this is that at lower threshold levels, the distinction between the nocuous and innocuous ambiguity starts to be obscure, which cause some difficulties for the ML-based approach.

VII. Threats to Validity

This section describes the potential threats that might affect the validity of our work and the nocuous ambiguities identified by our tool, and discuss how they are mitigated or accommodated.

Context Information. As mentioned earlier, anaphora ambiguity is a somewhat context-dependent ambiguity in which contextual information (e.g., discourse focus, the header of the text) provides some useful clues in interpreting the sentence. Structured documents (e.g., HTML and XML files) denote structural semantics for text such as title, headings, paragraphs, and other items. Unfortunately, quite a number of requirements documents are in PDF or txt format, which cannot provide this kind of useful structure information for analysis. Moreover, the relationships between requirements (e.g., parent and child requirements) which could also be utilized when making judgments.

In our surveys, judges are presented only with individual anaphora instances without the surrounding context. This can lead to additional difficulties for the judges in deciding on the correct interpretation. For example,

E13. *LVL1 trigger accept Identifier, A 24 bit L1ID is provided from the TTCrx with each LVL1A signal. In conjunction to the BCID,* ***it*** *uniquely defines an event.*

In this example, more than a half of the judges (i.e. 7 out of 13 judges) could not decide which NP candidate is most likely to be the antecedent of the anaphora with respect to the five NP candidates that appear in the preceding text. Therefore, for most thresholds, this instance exhibits nocuous ambiguity. However, the header information of this paragraph was 'L1ID'. Had the judges been provide with this, it is possible that this would have affected their choice of antecedent.

Coreference. Coreference occurs when multiple expressions in a sentence or document refer to the same entity in the world. Consider the instance (E14)

E14. *The Raw Data Processing Subsystem first checks the CCSDS parameters for valid values and stores these parameters if* ***they*** *are valid.*

This is a typical coreference case: since we know that the two NPs '*the CCSDS parameter*' and '*these parameters*' are most likely referring to the same parameters (i.e. they are coreferent). In our surveys, both of these NPs were separately selected as possible references of the pronoun '*they*' by our judges. Due to the lack of a coreference resolution capability in our tool, we treat these two NPs as unrelated NPs, and mistakenly recognize this anaphora ambiguity as *nocuous*. In practice, for many thresholds where this instance is classified as *innocuous* when it is in fact innocuous because the judgments for the two NPs should be combined. In future work, we intend to avoid such errors by adding an NP coreference resolution function, and allowing judges to select multiple possible antecedents.

TABLE VIII. The performance comarison between two ML-based models and the baseline model at different ambiguity thresholds t where weak positive threshold (W_Y=0.5) and weak negative threshold (W_N=0.4)

	ML-based Model (with weak thresholds)			ML-based Model (without weak thresholds)			Baseline Model (baseline)		
	P	R	F2	P	R	F2	P	R	F2
τ = 0.6	0.3006	0.9250	0.3921	0.3164	0.6500	0.3221	0.2000	1.0	0.3333
τ = 0.7	0.4324	0.9206	0.4506	0.4442	0.7460	0.3941	0.3150	1.0	0.4181
τ = 0.8	0.5595	0.9785	0.5106	0.5798	0.8602	0.4706	0.4650	1.0	0.4877
τ = 0.9	0.6786	0.9756	0.5382	0.6806	0.8780	0.4979	0.6150	1.0	0.5332
τ = 1.0	0.8220	0.9874	0.5695	0.8212	0.9245	0.5411	0.7950	1.0	0.5705

Machine Learning (ML) Approaches. To select an appropriate machine learning algorithm to build our antecedent classifier, we tested our dataset on a number of ML algorithms available in the WEKA package. The reason we used the Naive Bayes algorithm in our tool was because the algorithm generally performed better than other candidates at different levels of ambiguity thresholds, which achieved an average accuracy of 73.6% compared with Decision tree (70.39%), J48 (71.4%), LogitBoost (72.09%), and SVM (70.16%). However, during training, we found that the performance of this dataset on a variety of ML algorithms was relatively stable. Most of the ML algorithms achieve similar performance with the accuracy of above 70%. Still, it would be interesting to investigate whether the performance of our classifier can be improved by using other ML approaches such as Neural Networks or Genetic algorithms.

Other External Factors. In addition, there are a few external factors that are also considered to be threats to validity. For example, the requirements documents used in our dataset might not representative, or not cover various formats of texts; the distribution of the human judges might not representative in terms of native English speakers and background knowledge necessary to the judgments.

VIII. Related Work

Ambiguity in NL requirements. Several studies dealing with ambiguity identification have aimed to improve the quality of NL requirements documents. Some tools have been developed specifically to detect, measure, or reduce possible structural ambiguities in text. Kamsties, Berry, and Paech [13] describe a pattern-driven inspection technique to detect ambiguities in NL requirements. Fuchs and Swhwitter [12] present a restricted NL, called Attempt Controlled English (ACE), to translate specifications into sentences in first-order logic in order to reduce the ambiguity in requirement specifications. Mich and Garigliano [16] investigate the use of a set of ambiguity indices for the measurement in syntactic and semantic ambiguity, which is implemented using an NLP system called LOLITA. Kiyavitskaya, Zeni, Mich, and Berry [14] proposed a two-step approach in which a set of lexical and syntactic ambiguity measures are firstly applied to ambiguity identification, and then a tool to measure what specifically is potentially ambiguous about each sentence.

Finally, some of the tools were developed to examine the quality evaluation of requirements. These include QuARS [10], ARM [23], and Fantechi et al.'s tool [11] for use case requirements. These approaches define a quality model composed of a set of quality metrics (e.g., vagueness, subjectivity, optionality, weakness, etc.), and develop analysis techniques based on a linguistic approach to detect the defects related to the inherent ambiguity in the requirements.

Anaphora Resolution. Approaches to the anaphora resolution problem also vary. Previous research efforts have generally focused on either traditional linguistic methods, with the help of various types of linguistic and domain knowledge [15, 18], or statistics/corpus approach based on co-occurrence patterns [8], or machine learning approaches such as decision trees and SVM [1, 7, 20], or knowledge-poor approaches [19]. In addition, a number of research projects have applied the anaphora resolution techniques in various kinds of applications such as machine translation [21].

Similarly to other anaphora resolution approaches, we have explored a number of antecedent preferences to discover the preference for the candidates to the anaphor. However, our tool differs from the related work in that we have attempted to determine the degree to which an ambiguity is likely to be misinterpreted (i.e. whether it is nocuous relative to a particular ambiguity threshold), rather than attempting to resolve the anaphora resolution by applying disambiguation techniques to select the most like candidate as the antecedent of the anaphor.

IX. Conclusions and Future Work

Since many requirements documents continue to be written in natural language, we need ways to deal with the ambiguity inherent in natural language. Our overall research goal is to develop applications to detect potential nocuous ambiguity in requirements in order to minimize its effects.

In this paper, we presented an approach to automatically identify potentially nocuous ambiguities in requirements documents which contain anaphoric ambiguity. A set of anaphora ambiguities were extracted from a range of requirements documents, and associated human judgments on their interpretations were collected. We constructed a machine learning based classifier to predict the antecedent preference of noun phrase antecedent candidates. The antecedent information was then used to identify nocuous ambiguity. Our experimental results show that our tool achieves high recall with a consistent improvement on baseline precision subject to appropriate ambiguity thresholds, allowing us to highlight realistic and potentially problematic ambiguities in actual requirements documents.

Although based on significant technical development and substantive empirical studies, we believe that the application of our approach is actually lightweight and usable in that it allows requirements analysts to experiment and iterate to identify potential nocuous ambiguity in requirements, depending on their chosen analysis sensitivity threshold.

Nevertheless, there is ample room for improvement on a number of levels. For predication accuracy of nocuous ambiguity, a larger dataset is required in order to obtain more training instances for the construction of the antecedent classifier. One of the problems for our current classifier is the shortage of *positive* instances compared with *questionable* and *negative* instances. We expect that more positive instances would enhance the accuracy of the classifier. More research is also needed to exploit additional antecedent preferences that account for more aspects of anaphora.

In earlier work [24], we presented a methodology for the identification of nocuous ambiguity, which described three basic ideas underpinning our model of ambiguity: collection of human judgments, the heuristics used to model human interpretations, and a machine learning module to train the heuristics. This methodology has now been used effectively to handle coordination ambiguity [6, 22, 25] and anaphora ambiguity. We intend to apply the methodology to a wider range of ambiguity types, such as scope and preposition am-

biguity, which also contain rich syntactic and semantic information that can be modeled by various heuristics.

Finally, we envision our automated support for ambiguity analysis to fit into one of a number of requirements management environments, in which requirements authors are able to invoke our analysis tool in much the same way as writers invoke spell checkers. We are currently investigating the development of this capability within a well know commercial tool.

ACKNOWLEDGMENT

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) as part of the MaTREx project (EP/F068859/1), and by the Science Foundation Ireland (SFI grant 03/CE2/I303_1). We are grateful to our research partners at Lancaster University for their input, and to Ian Alexander for his practical insights and guidance.

REFERENCES

[1] C. Aone and S. W. Bennet, "Applying machine learning to anaphora resolution," in Connectionist, Statistical and symbolic approaches to learning for natural language processing, 1996, pp. 302-314.

[2] D. M. Berry, E. Kamsties, and M. M. Krieger, From contract drafting to software specification: Linguistic sources of ambiguity, 2003. A Handbook.

[3] D. Berry and E. Kamsties, "The syntactically dangerous all and plural in specifications" IEEE Software, vol. 22, pp. 55-57, 2005.

[4] S. Boyd, D. Zowghi, and A. Farroukh, "Measuring the expressiveness of a constrained natural language: an empirical study," Proc. the 13th IEEE International Conference on Requirements Engineering (RE'05), Washington, DC, 2005, pp. 339-352.

[5] S. E. Brennan, M. W. Friedman, and C. Pollard, "A centering approach to pronouns," Proc. the 25th Annual Meeting of the Association for Computational Linguistics (ACL), 1987, pp. 155–162.

[6] F. Chantree, B. Nuseibeh, A. D. Roeck, and A. Willis, "Identifying nocuous ambiguities in natural language requirements," Proc. 14th IEEE International Requirements Engineering Conference (RE'06), Minneapolis, USA, 2006, pp. 59-68.

[7] M. Connoly, J. D. Burger, and D. S. Day, "A machine learning approach to anaphoric reference," Proc. the International Conference on New Methods in Language Processing, 1994, pp. 255-261.

[8] I. Dagan and A. Itai, "Automatic processing of large corpora for the resolution of anaphora references," Proc. the 13th International Conference on Computational Linguistics (COLING'90) 1990, pp. 1-3.

[9] M. Denber, "Automatic resolution of anaphora in English," Eastman Kodak Co. Technical report 1998.

[10] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "The linguistic approach to the natural language requirements, quality: benefits of the use of an automatic tool," Proc. the twenty sixth annual IEEE computer society—NASA GSFC software engineering workshop, 2001, pp. 97–105.

[11] A. Fantechi, S. Gnesi, G. Lami, and A. Maccari, "Applications of Linguistic Techniques for Use Case Analysis," Requirements Engineering, vol. 8, pp. 161-170 2003.

[12] N. E. Fuchs and R. Schwitter, "Specifying logic programs in controlled natural language," Proc. the Workshop on Computational Logic for Natural Language Processing, 1995, pp. 3–5.

[13] E. Kamsties, D. Berry, and B. Paech, "Detecting ambiguities in requirements documents using inspections," Proc. the First Workshop on Inspection in Software Engineering (WISE'01), 2001, pp. 68-80.

[14] N. Kiyavitskaya, N. Zeni, L. Mich, and D. M. Berry, "Requirements for tools for ambiguity identification and measurement in natural language requirements specifications," Requirements Engineering Journal vol. 13, pp. 207–240 2008.

[15] S. Lappin and H. Leass, "An algorithm for pronominal anaphora resolution," Computational Linguistics, pp. 535-561, 1994.

[16] L. Mich and R. Garigliano, "Ambiguity measures in requirement engineering," Proc. international conference on software—theory and practice (ICS2000), 2000, pp. 39–48.

[17] L. Mich, M. Franch, and P. N. Inverardi, "Market research for requirements analysis using linguistic tools," Requirements Engineering Journal vol. 9, pp. 40–56, 2004.

[18] R. Mitkov, "An integrated model fro anaphora resolution," Proc. the 15th International Conference on Computational Linguistics (COLING'04), 1994, pp. 170-176.

[19] R. Mitkov, "Robust pronoun resolution with limited knowledge," Proc. the 18th International Conference on Computational Linguistics (COLING'98)/ACL'98 Montreal Canada 1998, pp. 869-875.

[20] V. Ng and C. Cardie, "Improving Machine Learning Approaches to Coreference Resolution," Proc. the 40th Annual Meeting of the Association for Computational Linguistics, 2002.

[21] H. Saggion and A. Carvalho, "Anaphora resolution in a machine translation system," Proc. the International Conference on Machine Translation, 1994, pp. 1-14.

[22] A. Willis, F. Chantree, and A. D. Roeck, "Automatic identification of nocuous ambiguity," Research on Language & Computation, vol. 6, pp. 1-23, 2008.

[23] W. M. Wilson, L. H. Rosenberg, and L. E. Hyatt, "Automated analysis of requirement specifications," Proc. the Nineteenth International Conference on Software Engineering (ICSE), 1997, pp. 161–171.

[24] H. Yang, A. d. Roeck, A. Willis, and B. Nuseibeh, "A Methodology for Automatic Identification of Nocuous Ambiguity," Proc. The 23th International Conference on Computational Linguistics (Coling'10), 2010. (In Press)

[25] H. Yang, A. Willis, A. D. Roeck, and B. Nuseibeh, "Automatic Detection of Nocuous Coordination Ambiguities in Natural Language Requirements," Proc. The 25th IEEE/ACM International Conference on Automated Software Engineering (ASE'2010), 2010. (In press)