Quality Factor Assessment and Text Summarization of Unambiguous Natural Language Requirements

R. Subha[1] and S. Palaniswami[2]

[1] Department of Computer Science and Engineering,
Sri Krishna College of Technology, Coimbatore – 641042, India
kris.subha@gmail.com
[2] Government College of Engineering, Bargur -635104, India

Abstract. The software requirements are documented in natural language to make it easy for the users to understand the document. This flexibility of natural language comes with the risk of introducing unwanted ambiguities in the requirements thus leading to poor quality. In this paper, we propose and evaluate a framework that automatically analyses the ambiguities in a requirements document, summarizes the document and assess its quality. We analyse the ambiguities that can occur in natural language and present a method to automate ambiguity analysis and consistency and completeness verification that are usually carried out by human reviewers which is time consuming and ineffective. The Open Text Summarizer based system summarizes the document and provides an extract of it. We use a decision tree based quality evaluator that identifies the quality indicators in the requirements document and evaluates it.

Keywords: Software Requirements Document, Natural language Processing, Ambiguity, Text summarization, Quality factors.

1 Introduction

Requirements collection plays a significant role during the development of a project. Requirements engineering is the process of discovering, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication and subsequent implementation. As a software development life cycle progresses, if the requirements retrieved are not formally documented or analyzed or updated, the quality of the software will degrade. Requirements consistency checking, Requirements tracing etc are done to improve the quality of the software. Most of these activities are done automatically and the rest are constituted by the human analyst. Therefore, to improve the quality of the software researchers are immensely involved in discovering better solutions. Ambiguity is an essential feature to be considered as it affects the Natural language requirements document and thereby affects the quality of the software. Many pre-processing activities are involved to carry out the ambiguity detection and classification. Earlier, the requirements gathering team was equipped with a handbook in order to remove ambiguity while preparing the requirements documents. When it comes to reading the requirements documents, they are large in amount and time consuming. The Natural Language requirements documents are error-prone. Before analyzing the requirements documents for the levels of ambiguity the evaluation of requirements documents needs to be considered as it involves a significant role in analyzing the document characteristics. A Software Requirements document (SRS) is unambiguous if, and only if, every requirement stated therein has only one interpretation", as stated in IEEE Recommended Practice for Software Requirements Specifications. SRSs are usually written in natural language, often augmented or enhanced by information in other notations, such as formulae, and diagrams. Natural Language is preferred to write every initial conceptual document and every request to proposal virtually. A recent online survey of businesses requiring software, conducted at University of Trento in Italy shows that a majority of documents available for requirements analysis are provided by the user or are obtained by interviews [2]. Moreover,

- 71.8% of these documents are written in common natural language,
- 15.9% of these documents are written in structured natural language, and
- Only 5.3% of these documents are written in formalized language.

1.1 Natural Language Processing

Natural Language Processing (NLP) is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. Automatic summarization, Co reference resolution, Morphological segmentation, Named entity recognition, Natural language generation, Natural language understanding, Part-of-speech tagging, Parsing, Relationship extraction, Sentence breaking, Sentiment analysis, Word segmentation, Word sense disambiguation are the tasks involved in NLP.

1.2 Text Summarization

A summary can be defined as a text that is produced from one or more texts, that contain a significant portion of the information in the original text(s), and that is no longer than half of the original text(s) [3][4]. Text summarization is the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks).When the summary replaces the original document; the output may be extract or abstract. A differentiation can be made between the Generic summaries and user-focused summaries (query-driven). Based on the output, a detailed differentiation is made between indicative summaries that indicate what topics are addressed in the source text and the details of what the original text is about, and the informative summaries, which are intended to cover the topics in the source text.

1.3 Quality Indicators

Quality Indicators [5] are syntactic aspects of the requirements specifications that can be automatically calculated and that provide information on a particular quality property of the requirements specifications themselves. The Quality indicators incorporated in the Quality model are classified into indicators related to Requirement Sentences Quality (RSQ) and Requirement Document Quality (RDQ). Requirement Sentences Quality (RSQ) related indicators are Implicit Subject Sentences, Multiple Sentences, Optional Sentences, Subjective Sentences, Underspecified Sentences, Vague Sentences and Weak Sentences. Requirements Document Quality (RDQ) related indicators are Comment frequency, Readability Index, under referenced Sentences and Unexplained Sentences.

2 Related Work

Wee Meng Soon et.al, proposed “A machine learning approach to co reference resolution of noun phrases” [12]. Here, a prerequisite for co reference resolution is to obtain the majority, if not all, of the possible markables[11][12] in a raw input text. To determine the markables, a pipeline of natural language processing (NLP) modules is used. It consists of tokenization, sentence segmentation, morphological processing, part-of-speech tagging, and noun phrase identification, named entity recognition. As far as co reference resolution is concerned, the goal of these NLP modules is to determine the periphery of the markables, and to endow with the indispensable information about each markable for subsequent generation of features in the training examples.

Chinatsu Aone and Scott William Bennett proposed “Applying Machine Learning to Anaphora Resolution”[1].This system uses feature vectors for pairs of an anaphor[6][9] and its possible antecedent. A total of 66 features are used, and they include lexical (e.g. category), syntactic (e.g. grammatical role), semantic (e.g. semantic class), and positional (e.g. distance between anaphor and antecedent) features. Those features can be either unary features (i.e. features of either an anaphor or an antecedent such as syntactic number values) or binary features (i.e. features concerning relations between the pairs such as the positional relation between an anaphor and an antecedent).

Elena Lloret proposed “Text Summarization: An overview”[4] which gives an overall idea about text summarization.Traditionally, summarization has been decomposed into three main stages[3][8] .

According to the Sparck Jones[8] approach, the stages are:

- Interpretation of the source text to obtain a text representation,
- Transformation of the text representation into a summary representation, and,
- Finally, generation of the summary text from the summary representation

Sparck Jones[8] distinguishes three classes of context factors:

- **Input factors.** The features of the text to be summarized crucially determine the way a summary can be obtained. These falls into three groups, which are: text form (e.g. document structure); subject type (ordinary, specialized or restricted) and unit (single or multiple) documents as input.

- **Purpose factors.** These are the most important factors. They fall under three categories: situation refers to the context within the summary is to be used; audience (i.e. summary readers) and use (what is the summary for?).
- **Output factors.** In this class, material (i.e. content), format and style, are grouped.

Hui Yang et.al, [6] developed an architecture of an automated system to support requirements writing, by incorporating nocuous ambiguity detection into the requirements workflow. The core of such architecture comprises a classifier that automatically determines whether an instance of anaphoric ambiguity is nocuous or innocuous. The classifier is developed using instances of anaphoric ambiguity extracted from a collection of requirements documents. For each instance, a set of human judgments are used to classify. A classifier is then trained on the linguistic features of the text and the distribution of judgments to identify instances of nocuous ambiguity in new cases. Several approaches can be followed to ensure a good quality requirements document. Another approach is the linguistic analysis of a NL requirements document intended to confiscate most of the issues related to readability and ambiguity. A lot of studies dealing with the evaluation and the achievement of quality in NL requirement documents can be found in the literature and Natural Language Processing (NLP) tools have been recently applied to NL requirements documents for inspecting the consistency and completeness.

3 Methodology

In this paper, we propose a method for summarizing quality requirements which includes an open NLP based system using the MaxEnt models for the detection of sentence end words, tokens, parts-of-speech, named entities, anaphors and co referring phrases. The ambiguity detection module of the system is built on the model proposed by Hui Yang et.al [6] and is refined to reduce human intervention. The detection of anaphoric ambiguity and identification of the co-referring noun phrases based on the anaphors and their relationship with other words are done automatically by the system. The unambiguous requirements are summarized to the number of sentences specified by the user. The system has the quality evaluation module to evaluate the quality of the NL Requirements document. The proposed system takes requirements document [15] as input and reads the content. The sentence boundaries are detected by the sentence detector by reading the contents of the file. The tokenize module gets the input which is in the form of sentences and identifies the tokens in the sentences. These tokens are subsequently used by the POS tagger to mark the Parts-of-Speech tags. The construction of parse tree is done by the parser by utilizing the POS tag details. The parse tree is used by the ambiguity detection module which uses a classifier both to identify the co referring Noun Phrases (NP) present in the sentences as well as to classify the sentences as ambiguous. The significant sentences are extracted by the text summarizer from the document. The contents of the requirements document are utilized in order to discover the quality indicators and evaluate them by the quality evaluator.

3.1 Text Preprocessing

The critical part of any NLP system is the Text pre-processing since the characters, words, and sentences recognized at this stage constitute the elemental units passed to all advanced processing stages, from analysis and tagging components, such as morphological analyzers and part-of-speech taggers, through applications, such as information retrieval and machine translation systems. A “pipeline” of text processing components is used in order to provide value from text by the NLP applications, such as customizable information extraction or question answering application. By means of these systems, the performance of each successive system depends on the performance of each of the components that preceded it in the pipeline. In this way, errors made by an "upstream" component (like a part-of-speech tagging system) can cause a negative impact on the performance of each "downstream" system (such as a named entity recognizer or co reference resolution system).This is shown in Fig 1.

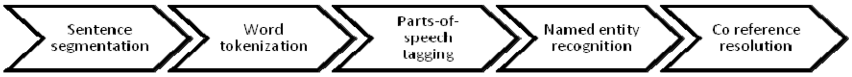


Fig. 1. Pipeline of Natural language processing modules

The modules of natural language processing can be described as follows.

Sentence Segmentation [14]. The documents are split into sentences by the Sentence Segmentation system that are later processed and annotated by "downstream" components. When scanned through the input text, one of these characters ('.', '!', '?') is encountered and a way for deciding whether or not it marks the end of a sentence is to be decided. Here maximum entropy model (MaxEnt) is used. A set of predicates related to the possible end-of-sentence positions is generated. Various features, relating to the characters before and after the possible end-of-sentence markers, are used to generate this set of predicates. This set of predicates is then evaluated against the MaxEnt model. The characters including the position of the end-of-sentence marker are separated off into a new sentence if the best outcome indicates a sentence break. The indication whether a punctuation character denotes the end of a sentence or not is detected by the Sentence Detector. This shows that a sentence is defined as the longest white space trimmed character sequence between two punctuation marks. An exception to this rule would be the first and last sentence. The first non whitespace character is assumed to be the beginning of a sentence, and the last non whitespace character is assumed to be an ending of a sentence. It is also possible to perform tokenization first and let the Sentence Detector process the already tokenized text but usually Sentence Detection is done before the text is tokenized thereby using the pre-trained models.

Tokenization [14]. Tokenization systems break sentences into sets of word-like objects which represent the smallest unit of linguistic meaning considered by a natural language processing system. This tokenize module will split words that comprises of contractions: for example, it will split "don't" into "do" and "n't", because it is designed to pass these tokens on to the other NLP tools, where "do" is recognized as a verb, and "n't" as a contraction of "not", an adverb modifying the preceding verb "do".

The input character sequences are split into tokens by the Tokenize module. Tokens are usually words, punctuation, numbers, etc. It is essential to ensure that tokenization produces tokens of the type expected by later text processing components.

Various tokenizer implementations are

- Whitespace Tokenizer - A whitespace tokenizer, non whitespace sequences are identified as tokens
- Simple Tokenizer - A character class tokenizer, sequences of the same character class are tokens
- Learnable Tokenizer - A maximum entropy tokenizer, detects token boundaries based on probability model

Parts-of-Speech tagging [14]. Part-of-speech tagging is the act of assigning a Part of Speech (POS) to each word in a sentence. The POS tags consist of coded abbreviations conforming to the scheme of the Penn Treebank, the linguistic corpus developed by the University of Pennsylvania. Based on the token itself and the context of the token the Part of Speech Tagger marks tokens with its consequent word type. A token can have multiple POS tags depending on the token and the context. The POS Tagger uses a probability model to identify the correct POS tag out of the tag set. A tag dictionary is used to increase the tagging and runtime performance of the tagger in order to restrict the possible tags for a token.

Named Entity Recognition[14]. Named Entity Recognition systems categorize phrases (referred to as entities) found in text with respect to a potentially large number of semantic categories, such as person, organization, or geopolitical location."Name finding" is the term used by the OpenNLP library to refer to the identification of classes of entities within the sentence - for example, people's names, locations, dates, and so on. Seven different types of entities, symbolized by the seven maximum entropy model files in the NameFind subfolder - date, location, money, organization, percentage, person, and time are established by the name finder .Other classes of entities are set up by utilizing the SharpEntropy library to train the new models. The algorithm is far from foolproof as it is dependent on the use of training data and there are many, many tokens that might come into a category such as "person" or "location".

NLP Models. OpenNLP[14] models are trained models developed for use in the NLP. The list of models used in the system is shown in Table 1.

Table 1. List of Models

Component	Description
Tokenizer	Trained on OPENNLP training data
Sentence Detector	Trained on OPENNLP training data
POS Tagger	MAXENT model with tag dictionary
Name Finder	Date name finder model
Name Finder	Location name finder model
Name Finder	Money name finder model
Name Finder	Organization name finder model
Name Finder	Percentage name finder model
Name Finder	Person name finder model
Name Finder	Time name finder model
Chunker	Trained on conll2000 shared task data

3.2 Ambiguity Detection

The ambigituy detection module includes coreference resolution and ambiguity classification.

Co Reference Resolution. Co reference Resolution is the process of identifying the linguistic expressions which make reference to the same entity or individual within a single document or across a collection of documents. Co reference occurs when multiple expressions in a sentence or document refer to the same entity in the world. Initially all possible references need to be extracted from the document before determining the co reference for a document. Every reference is a possible anaphor, and every reference before the anaphor in document order is a possible antecedent of the anaphor, except when the anaphor is nested. If the anaphor is a child or nested reference, then the possible antecedents must not be any reference with the same root reference as the current anaphor [11][12]. Still, the possible antecedents can be other root references and their children that are before the anaphor in document order. The new ambiguous instance, potential pairs of co referring NPs are offered to the classifier to resolve whether the two NPs co refer or not in order to estimate the co reference relations among the possible NPs antecedent candidates. In this system, heuristics-based methods are built-in to exploit the factors that influence co reference determination. The heuristics are incorporated in terms of feature vectors and are modeled based on the Table 2.

Table 2. Feature vector description for coreference resolution heuristics

Feature type	Feature	Description
String matching	Full string matching	Y if both NPs contain the same string aftere the removal of non-informative words,else N
	Head word matching	Y if both NPs contain the same Headword,else N
	Modifier matching	Y if both NPs share the same modifier substring, else N
	Alias name	Y if one NP is the alias name of the other NP, else N
Grammatical	NP type (NP_i)	Y if NP_i is either definite NP or demonstrative NP, else N
	NP type (NP_j)	Y if NP_j is either definite NP or demonstrative NP, else N
	Proper name	Y if both NPs are proper names, else N
	Number agreement	Y if NP_i and NP_j agree in number, else N
Syntactic	PP attachment	Y if one NP is the PP attachment of the other NP, else N
	Appostive	Y if one NP is in appostive to the other NP, else N
	Syntactic role	Y if both NPs have the same syntactic role in the sentence, else N

Each instance of an anaphor is associated with a set of candidate antecedents. A pair wise comparison of the NPs is accomplished by the classifier to identify potential co reference relations among the candidate antecedents. Likewise, each NP pair is tested for co reference, and sets of co referent candidates are identified.

Ambiguity Classification. Anaphoric ambiguity [7] occurs when the text offers two or more potential antecedent candidates either in the same sentence or in a preceding one, as in, 'The function shall build the parse tree, and then display it in a new window'. The expression to which an anaphor [9] refers is called its antecedent. Antecedents for personal pronoun [10] anaphora are nouns or noun phrases (NPs) found elsewhere in the text, usually preceding the anaphor itself. Based on multiple human judgments of the suitable NP antecedent candidate in terms of an anaphoric ambiguity instance [6], the antecedent can be classified. A number of preference heuristics are also included to model the factors that may favor a particular interpretation. A machine learning algorithm is implemented with a set of training instances to construct a classifier. Given an anaphor and a set of possible NP antecedents, the classifier then predicts how strong the preference for each NP is, and from there, whether the ambiguity is nocuous or innocuous. The Naive Bayes classifier is used to classify the antecedents.

Naive Bayes Classification
The Naive Bayes Classifier technique is based on the Bayesian theorem and is particularly suited when the dimensionality of the inputs is high.

Algorithm
```
D : Set of tuples
    Each Tuple is an 'n' dimensional attribute vector
    X : (x1,x2,x3,.... xn)
    Let there be 'm' Classes : C1,C2,C3...Cm
    Naive Bayes classifier predicts X belongs to Class Ci iff
    $P(C_i/X) > P(C_j/X)$ for $1<= j <= m$ , $j <> i$
    Maximum Posteriori Hypothesis
    $P(C_i/X) = P(X/C_i) P(C_i) / P(X)$
    Maximize $P(X/C_i) P(C_i)$ as $P(X)$ is constant
with many attributes, it is computationally expensive to evaluate $P(X/C_i)$.
    Naive's Assumption of "class conditional independence"
    $P(X/C_i) = P(x_1/C_i) * P(x_2/C_i) *...* P(x_n/ C_i)$
```

The Naive bayes classifier uses the feature vectors in Table 3 to classify the antecedent and the anaphoric ambiguity.

Table 3. Feature vector description for Antecedent classification heuristics

Feature Type	Feature	Description
Linguistics	Number agreement	Y if NP agree in number; N_P if NP does not agree in number but it has a person property; N if NP doesn't agree in number;UNKNOWN if the number information cannot be determined
	Definiteness	Y if NP is a definite NP; else N Non-prepositional NP Y if NP is a non-prepositional NP; else N
	Syntactic constraint	Y if NP satisfies syntactic constraint; else N
	Syntactic parallelism	Y if NP satisfies syntactic parallelism; else N
	Coordination pattern	Y if NP satisfies coordination pattern; else N
	Non-associated NP	Y if NP is a non-associated NP; else N
	Indicating verb	Y if NP follows one of the indicating verbs; else N
	Semantic constraint	Y if NP satisfies semantic constraint; else N
	Semantic parallelism	Y if NP satisfies semantic parallelism; else N
	Domain-specific term	Y if NP is contained in the domain-specific term list; else N
Context	Centering	Y if NP occurs in the paragraph more than twice; else N
	Section heading	Y if NP occurs in the heading of the section; else N
	Sentence recency	INTRA_S if NP occurs in the same sentence as the anaphor; else INTER_S
	Proximal	Integral value n, where n means that NP is the nth 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 in local requirements document
	BNC-based collocation frequency	Y if the matched co-occurrence pattern containing NP appears in the word list returned by the sketch engine; else N

The machine learning algorithm allots a weighted antecedent tag to the NP candidate while they are presented with a pronoun and a candidate. Likewise, the antecedent tag information is used by the system to predict whether the anaphora instance displays nocuous ambiguity and then they are disambiguated.

3.3 Text Summarization

Summarization can be exemplified as approaching the problem at the surface, entity, or discourse levels in the conventional system. In this paper, system surface level approach is used. Surface level approach inclines to represent information taking shallow features and then selectively combining them together in order to obtain a salience function that can be used to extract information.Among these features, are:

Thematic features rely on word (significant words) occurence statistics, so that sentences containing words that occur frequently in a text have higher weight than the rest which illustrates the fact that these sentences are the vital ones and they are hence extracted. Before doing term frequency, altering task must be done using a stop-list words which contains words such as pronouns, prepositions and articles. This is the classical statistical approach. Nevertheless, from a point of view of a corpus-based approach td*idf measure (commonly used in information retrieval) is extremely useful to determine keywords in text.

Location refers to the position in text, paragraph or any other particular section which exhibits the point that they contain the target sentences to be included in the summary. This is usually genre-dependent, but there are two fundamental wide-ranging methods, namely leadmethod and the title-based method with cue-word method .

Background assumes that the importance of meaning units is detemined by the presence of terms from the title or headings, initial part of the text or a user's query.

Cue words and phrases, such as "in conclusion", "important", "in this paper",etc. can be very useful to determine signals of relevance or irrelevance and such units are detected both automatically and manually.The text summarization is implemented based on Open Text Summarizer.

3.4 Quality Factor Assessment

Quality indicators are syntactic aspects of the requirements specifications that can be automatically calculated and that provide information on a particular quality property of the requirements specifications themselves. The system uses decision tree to evaluate the quality of the NL requirement document.Table 4 shows the different quality indicators and their description.

Table 4. Quality indicators and their description

Quality indicator	Description	Notes
Optionality	An Optionality Indicator reveals a requirement sentence containing an optional part (i.e. a part that can or cannot considered)	Optionality- revealing words: possibly, eventually, if case,if possible, if appropriate, if needed
Subjectivity	A Subjectivity Indicator is pointed out if sentence refers to personal opinions or feeling	Subjectivity-revealing wordings: similar, better, similarly, worse, having in mind, take into account, take into consideration, as[adjective] as possible
Vagueness	A Vagueness Indicator is pointed out if the sentence includes words holding inherent vagueness, i.e. words having a non uniquely quantifiable meaning	Vagueness-revealing words:clear, easy, strong, good,bad, efficient, useful,significant, adequate, fast,recent, far, close, in front
Weakness	A Weakness Indicator is pointed out in a sentence then it contains a weak main verb	Weak verbs: can, could,may.

Table 4. (*Continued)*

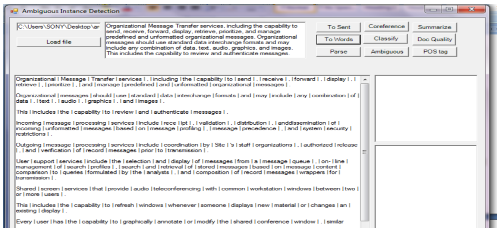
Implicity	An Implicity Indicator is pointed out in a sentence when the subject is generic rather than specific.	Subject expressed by: Demonstrative adjective (this, these,that, those) or Pronouns (it, they, ..). Subject specified by:Adjective previous,next, following, last,..)or Preposition
Readability	It is the value of ARI (Automated Readability Index) [ARI=WS + 9*SW where WS is the average words per sentence, SW is the average letters per word]	

4 Experiment and Results

The system is implemented using C#,.Net as front end and SQLite as the back end.

4.1 Text Preprocessing

Text preprocessing involves steps to be performed before the original text is being given as input to the next stage of the system. The requirements document written in natural language is given as the input and parsed sentences are the output. The Sentence Detector detects whether the punctuation character marks the end of a sentence or not.(i.e) Given a chunk of text, find the sentence boundaries. The input character sequence are broken into tokens.(i.e) Separate a chunk of continuous text into separate words by the Tokenizer. Tokens are usually words, punctuation, numbers, etc. Fig 2 shows the output of tokenizer.

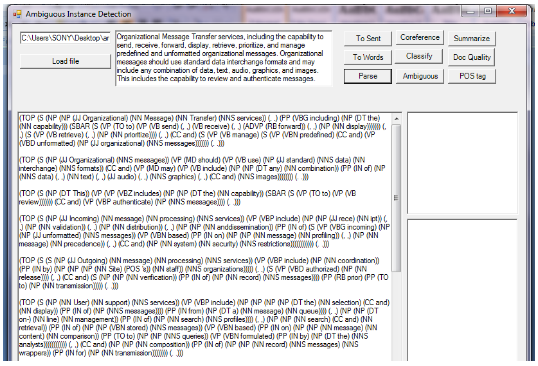


```
Organizational Message Transfer services, including the capability to
send, receive, forward, display, retrieve, prioritize, and manage
predefined and unformatted organizational messages. Organizational
messages should use standard data interchange formats and may
include any combination of data, text, audio, graphics, and images.
This includes the capability to review and authenticate messages.
```

```
Organizational | Message | Transfer | services | , | including | the | capability | to | send | , | receive | , | forward | , | display | , |
retrieve | , | prioritize | , | and | manage | predefined | and | unformatted | organizational | messages | .

Organizational | messages | should | use | standard | data | interchange | formats | and | may | include | any | combination | of |
data | , | text | , | audio | , | graphics | , | and | images | .

This | includes | the | capability | to | review | and | authenticate | messages | .

Incoming | message | processing | services | include | rece | ipt | , | validation | , | distribution | , | anddissemination | of |
incoming | unformatted | messages | based | on | message | profiling | , | message | precedence | , | and | system | security |
restrictions | .

Outgoing | message | processing | services | include | coordination | by | Site | 's | staff | organizations | , | authorized | release
| , | and | verification | of | record | messages | prior | to | transmission | .

User | support | services | include | the | selection | and | display | of | messages | from | a | message | queue | , | on- | line |
management | of | search | profiles | , | search | and | retrieval | of | stored | messages | based | on | message | content |
comparison | to | queries | formulated | by | the | analysts | , | and | composition | of | record | messages | wrappers | for |
transmission | .

Shared | screen | services | that | provide | audio | teleconferencing | with | common | workstation | windows | between | two |
or | more | users | .

This | includes | the | capability | to | refresh | windows | whenever | someone | displays | new | material | or | changes | an |
existing | display | .

Every | user | has | the | capability | to | graphically | annotate | or | modify | the | shared | conference | window | , | similar
```

Fig. 2. Tokenization

The Part of Speech Tagger in scripts tokens with their equivalent word type based on the token itself and the context of the token. A token might have multiple POS tags depending on the token and the context. A tag dictionary is employed to increase the tagging and runtime performance of the tagger in order to restrict the possible tags for a token. Parsing determines the parse tree (grammatical analysis) of a given sentence. The output of parsing is shown in Fig 3.



```
C:\Users\SONY\Desktop\ar
```

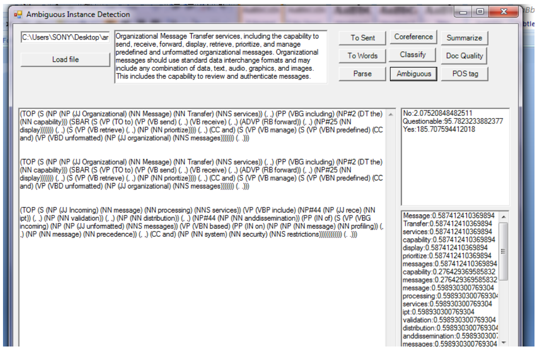
```
Organizational Message Transfer services, including the capability to
send, receive, forward, display, retrieve, prioritize, and manage
predefined and unformatted organizational messages. Organizational
messages should use standard data interchange formats and may
include any combination of data, text, audio, graphics, and images.
This includes the capability to review and authenticate messages.
```

```
(TOP (S (NP (NP (JJ Organizational) (NN Message) (NN Transfer) (NNS services)) (, ,) (PP (VBG including) (NP (DT the)
(NN capability))) (SBAR (S (VP (TO to) (VP (VB send) (, ,) (VB receive) (, ,) (ADVP (RB forward)) (, ,) (NP (NN display))))))) (,
,) (S (VP (VB retrieve) (, ,) (NP (NN prioritize)))) (, ,) (CC and) (S (VP (VB manage) (S (VP (VBN predefined) (CC and) (VP
(VBD unformatted) (NP (JJ organizational) (NNS messages))))))) (. .)))

(TOP (S (NP (JJ Organizational) (NNS messages)) (VP (MD should) (VP (VB use) (NP (JJ standard) (NNS data) (NN
interchange) (NNS formats)) (CC and) (VP (MD may) (VP (VB include) (NP (NP (DT any) (NN combination)) (PP (IN of) (NP
(NNS data) (, ,) (NN text) (, ,) (JJ audio) (, ,) (NNS graphics) (, ,) (CC and) (NNS images)))))))) (. .)))

(TOP (S (NP (DT This)) (VP (VP (VBZ includes) (NP (NP (DT the) (NN capability)) (SBAR (S (VP (TO to) (VP (VB
review))))))) (CC and) (VP (VBP authenticate) (NP (NNS messages)))) (. .)))

(TOP (S (NP (JJ Incoming) (NN message) (NN processing) (NNS services)) (VP (VBP include) (NP (NP (JJ rece) (NN ipt)) (,
,) (NP (NN validation)) (, ,) (NP (NN distribution)) (, ,) (NP (NP (NN anddissemination)) (PP (IN of) (S (VP (VBG incoming) (NP
(NP (JJ unformatted) (NNS messages)) (VP (VBN based) (PP (IN on) (NP (NP (NN message) (NN profiling)) (, ,) (NP (NN
message) (NN precedence)) (, ,) (CC and) (NP (NN system) (NN security) (NNS restrictions)))))))))))) (. .)))

(TOP (S (S (NP (JJ Outgoing) (NN message) (NN processing) (NNS services)) (VP (VBP include) (NP (NN coordination))
(PP (IN by) (NP (NP (NP (NN Site) (POS 's)) (NN staff)) (NNS organizations))))) (, ,) (S (VP (VBD authorized) (NP (NN
release)))) (, ,) (CC and) (S (NP (NP (NN verification)) (PP (IN of) (NP (NN record) (NNS messages)))) (PP (RB prior) (PP (TO
to) (NP (NN transmission))))) (. .)))

(TOP (S (NP (NN User) (NN support) (NNS services)) (VP (VBP include) (NP (NP (NP (NP (DT the) (NN selection) (CC and)
(NN display)) (PP (IN of) (NP (NNS messages)))) (PP (IN from) (NP (DT a) (NN message) (NN queue)))) (, ,) (NP (NP (DT
on-) (NN line) (NN management)) (PP (IN of) (NP (NN search) (NNS profiles)))) (, ,) (NP (NP (NN search) (CC and) (NN
retrieval)) (PP (IN of) (NP (NP (VBN stored) (NNS messages)) (VP (VBN based) (PP (IN on) (NP (NP (NN message) (NN
content) (NN comparison)) (PP (TO to) (NP (NP (NNS queries)) (VP (VBN formulated) (PP (IN by) (NP (DT the) (NNS
analysts)))))))))))) (, ,) (CC and) (NP (NP (NN composition)) (PP (IN of) (NP (NP (NN record) (NNS messages) (NNS
wrappers)) (PP (IN for) (NP (NN transmission)))))))) (. .)))
```

Fig. 3. Parser

4.2 Ambiguity Detection

The parsed sentences from requirements document are given as the input and Co referring NPs are the output. Based on multiple human judgments of the most likely NP antecedent candidate in terms of an anaphoric ambiguity instance a classifier is implemented. For every pronoun and associated NP antecedent candidates, the antecedent classifier allots one of the antecedent preference labels, Positive (Y), Questionable (Q), and Negative (N), to each candidate. Later the calculation is done to verify whether the anaphoric ambiguity is nocuous or innocuous. This information is then used to Fig 4 shows the output of the ambiguity detection module. The requirements statements are classified based on the threshold value.



```
C:\Users\SONY\Desktop\ar
```

```
Organizational Message Transfer services, including the capability to
send, receive, forward, display, retrieve, prioritize, and manage
predefined and unformatted organizational messages. Organizational
messages should use standard data interchange formats and may
include any combination of data, text, audio, graphics, and images.
This includes the capability to review and authenticate messages.
```

```
(TOP (S (NP (NP (JJ Organizational) (NN Message) (NN Transfer) (NNS services)) (, ,) (PP (VBG including) (NP#2 (DT the)
(NN capability))) (SBAR (S (VP (TO to) (VP (VB send) (, ,) (VB receive) (, ,) (ADVP (RB forward)) (, ,) (NP#25 (NN
display))))))) (, ,) (S (VP (VB retrieve) (, ,) (NP (NN prioritize)))) (, ,) (CC and) (S (VP (VB manage) (S (VP (VBN predefined) (CC
and) (VP (VBD unformatted) (NP (JJ organizational) (NNS messages))))))) (. .)))


(TOP (S (NP (NP (JJ Organizational) (NN Message) (NN Transfer) (NNS services)) (, ,) (PP (VBG including) (NP#2 (DT the)
(NN capability))) (SBAR (S (VP (TO to) (VP (VB send) (, ,) (VB receive) (, ,) (ADVP (RB forward)) (, ,) (NP#25 (NN
display))))))) (, ,) (S (VP (VB retrieve) (, ,) (NP (NN prioritize)))) (, ,) (CC and) (S (VP (VB manage) (S (VP (VBN predefined) (CC
and) (VP (VBD unformatted) (NP (JJ organizational) (NNS messages))))))) (. .)))


(TOP (S (NP (JJ Incoming) (NN message) (NN processing) (NNS services)) (VP (VBP include) (NP#44 (NP (JJ rece) (NN
ipt)) (, ,) (NP (NN validation)) (, ,) (NP (NN distribution)) (, ,) (NP#44 (NP (NN anddissemination)) (PP (IN of) (S (VP (VBG
incoming) (NP (NP (JJ unformatted) (NNS messages)) (VP (VBN based) (PP (IN on) (NP (NP (NN message) (NN profiling)) (,
,) (NP (NN message) (NN precedence)) (, ,) (CC and) (NP (NN system) (NN security) (NNS restrictions)))))))))))) (. .)))
```

```
No:2.07520848482511
Questionable:95.7823233882377
Yes:185.707594412018
```

```
Message:0.587412410369894
Transfer:0.587412410369894
services:0.587412410369894
capability:0.587412410369894
display:0.587412410369894
prioritize:0.587412410369894
messages:0.587412410369894
capability:0.276429369585832
messages:0.276429369585832
message:0.598930300769304
processing:0.59893030076930
services:0.598930300769304
ipt:0.598930300769304
validation:0.598930300769304
distribution:0.59893030076930
anddissemination:0.59893030
messages:0.598930300769304
```

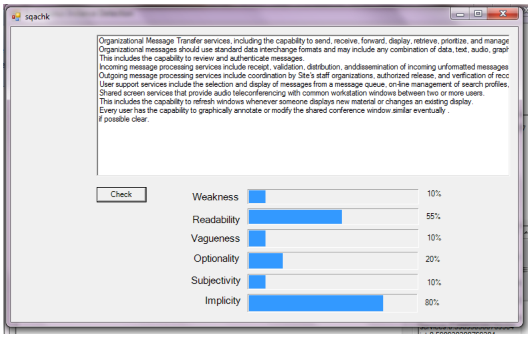
Fig. 4. Ambiguity Classification

4.3 Text Summarization

Text summarization involves reducing a text document or a larger corpus of multiple documents into a short set of words or paragraph that conveys the main meaning of the text. Sentence detection and tokenization,token ranking and finally prioritizing sentences based on the accumulated ranking are performed in Text Summarization.

4.4 Quality Factor Evaluation

The evaluation of the quality factors of the requirements document are carried out by involving qualtiy indicators.The document is verified and the quality values are assigned based on the quality indicators.This is shown in Fig 5.


Organizational Message Transfer services, including the capability to send, receive, forward, display, retrieve, prioritize, and manage
Organizational messages should use standard data interchange formats and may include any combination of data, text, audio, graph
This includes the capability to review and authenticate messages.
Incoming message processing services include receipt, validation, distribution, anddissemination of incoming unformatted messages
Outgoing message processing services include coordination by Site's staff organizations, authorized release, and verification of reco
User support services include the selection and display of messages from a message queue, on-line management of search profiles,
Shared screen services that provide audio teleconferencing with common workstation windows between two or more users.
This includes the capability to refresh windows whenever someone displays new material or changes an existing display.
Every user has the capability to graphically annotate or modify the shared conference window.similar eventually .
if possible clear.

Weakness	10%
Readability	55%
Vagueness	10%
Optionality	20%
Subjectivity	10%
Implicity	80%

Fig. 5. Quality Factor Evaluation

5 Performance Evaluation

The output of the Ambiguity detection module is compared with "ARKref NP coreference system"[16]. The ARKref coreference resolution system is implemented in java and is available in the web. The ARKref resolution system uses the BNC corpora, web corpora and Wordnet to identify the NP coreferences among the NPs in the sentences. The result of the comparison is shown in Table 5. The input for the evaluation are taken from the requirements dataset collected from RE@UTS website [15].

Table 5. Results of Coreference resolution

S.No	Number of input sentences	Actual NP coreference	NP Coreferences detected	
			Proposed system	ARKref system
1	11	3	3	2
2	10	8	7	7
3	10	6	6	4

Table 6. Precision, Recall, F Measure values

S.No	Proposed system			ARKref system		
	Precision	Recall	F Measure	Precision	Recall	F Measure
1	1	0.2727	0.4284	0.666	0.1818	0.2848
2	0.8757	0.7	0.777	0.8757	0.7	0.777
3	1	0.6	0.75	0.666	0.4	0.4998

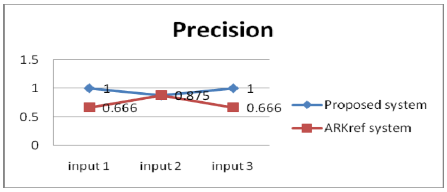


Fig. 6. Precision for co reference resolution

From the precision for co reference resolution graph (Fig 6) it is inferred that the proposed system provides accurate result over ARKref system.

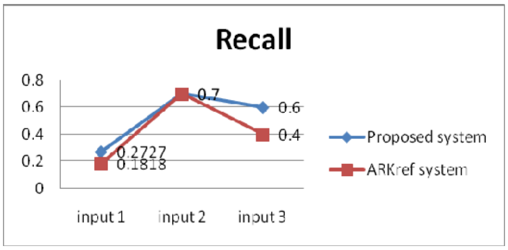


Fig. 7. Recall for co reference resolution

The Recall values in Fig 7 show that the proposed system provides false positives less than that of the ARKref system.

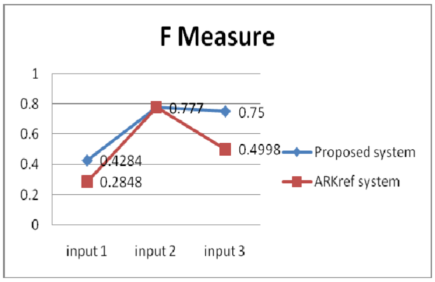


Fig. 8. F Measure for coreference resolution

Table 7 shows ambiguity judgement variation based on the threshold value set by the requirement analyst.

Table 7. Results of Ambiguity detection

Threshold value	Actual ambiguity	Detected nocuous ambiguity
0.5	3	1
0.6	3	2
0.65	3	3
0.7	3	3
0.75	3	3
0.8	3	3
0.85	3	6
0.9	3	7
1	3	8

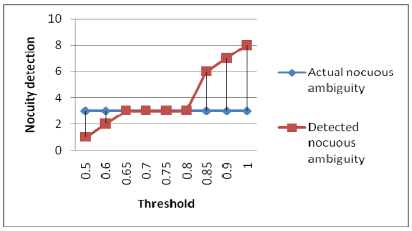


Fig. 9. Threshold vs Nocuous ambiguity detection

The graph in Fig 9 shows that the nocuity detection varies as the threshold value increases. The accuracy of nocuity detection drops when the threshold value is set above 0.8. Thus better accuracy is achieved in the proposed system when the threshold is set between 0.65 and 0.8.

6 Conclusion

Thus a framework is proposed that can automatically analyse the ambiguities for a given natural language requirement document. Initially the corefering NPs were utilised to classify the pronouns and further the ambiguity detection module was used to detect the nocuous statements from the requirments document by the antecedent classifier.Based on the threshold value the nocuous anaphoric ambiguity was differentiated from that of the innocuous ambiguities. The system used OpenNLP models developed by OpenNLP Apache software foundation. The accuracy of the proposed system was enhanced by Brown corpora and Wordnet dictionary. The Open Text Summarizer based system summarized the document and provided an extract of it. The decision tree based quality evaluator identified the quality indicators effectively in the requirements document and evaluated the quality of the requirement document provided for analysis.

References

1. Aone, C., Bennett, S.W.: Applying machine learning to anaphora resolution. In: Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing, pp. 302–314 (1996)
2. Berry, D.M., Kamsties, E., Krieger, M.M.: From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity, University of Waterloo, Ontario, Canada (2003)
3. Hovy, E.: Automated Text Summarization. In: Mitkov, R. (ed.) The Oxford Handbook of Computational Linguistics (2005)
4. Lloret, E., Palomar, M.: Text summarization in progress: a literature review. Artificial Intelligence Review 37 (2012)
5. Lami, G.: QuARS: A Tool for Analyzing Requirement, CMU/SEI-2005-TR-014 (2005)
6. Yang, H., de Roeck, A., Gervasi, V., Willis, A., Nuseibeh, B.: Analyzing anaphoric ambiguity in natural language requirement. Requirements Engineering Journal 16, 163–189 (2011)
7. Dagan, I., Itai, A.: Automatic processing of large corpora for the resolution of anaphora references. In: Proceedings of the 13th International Conference on Computational Linguistics, pp. 1–3 (1990)
8. Jones, K.S.: Automatic summarizing: factors and directions. Advances in Automatic Text Summarization. MIT Press (1999)
9. Denber, M.: Automatic resolution of anaphora in English. Technical report. Eastman Kodak Co. (1998)
10. Brennan, S.E., Froedman, M.W., Pollard, C.J.: A centering approach to pronouns. In: Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 155–162 (1987)
11. Ng, V., Cardie, C.: Improving machine learning approaches to co reference resolution. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 104–111 (2002)
12. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A machine learning approach to co reference resolution of noun phrases. Computational Linguistics - Special Issue on Computational Anaphora Resolution Archive 27, 521–544 (2001)
13. `http://incubator.apache.org/opennlp/`
14. `http://research.it.uts.edu.au/re/`
15. `http://www.ark.cs.cmu.edu/ARKref/`