

Proposal for Avoiding Ambiguity in Requirement Engineering using Artificial Intelligence

Shagun Sinha¹, Mohd Shahid Husain²

*Dept. of Computer Sc. & Engg.
Integral University, Lucknow, India*

Abstract- Ambiguity, the biggest problem in the Software Development Life Cycle and this proposal is to introduce an idea for avoiding the ambiguity during the time of Requirement Elicitation. While developing software many a times it is not clear that a particular word or the particular sentence has two or more meanings and if we know, even then it is difficult to decide which one is correct. Hence in this proposal I am sharing a concept of a tool which can resolve this problem. This tool will be able to identify the ambiguous words and provide all the possible meanings of those ambiguous words clarifying the meaning of the sentence. Amongst all the possible meanings of the ambiguous words the elicitor can choose the appropriate word even after discussing with the stake holder(s). Currently the artificial intelligence is not so advanced that we can write the program which can choose the correct meaning of those words itself. Human involvement is required in this because the points in this process are so important that we cannot rely completely on machines and programs.

Keywords- Ambiguity, WordNet, Part-Of-Speech Tagger, TAAI.

I. INTRODUCTION

This proposal is an attempt to avoid the ambiguity during requirement elicitation. Requirement Elicitation is very important part of the software development life cycle. Every phase of SDLC after requirement analysis depends upon Requirement Elicitation[1]. Requirement Elicitation is the process in which some information is collected, and this information can be used for the development of the software. The problem faced in this phase is the Ambiguity.

The collection of information is done through various sources and stake holders are one of them. They are no programmers but they are the one who are related to that organization or they have knowledge about the organization for which we are creating the software. Hence they are the big support in gathering the information, but the problem is that many times it is found that whatever is documented carries a large amount of ambiguity. Hence the ultimate product is not what is expected and not only that rectifying the error causes: wastage of time, extra money and extra efforts to track down the error and make changes.

Ambiguity means that any sentence or word with more than one meaning like the word 'saw' it has several meanings like 'see', 'cutting tool' and 'cut', hence it is a ambiguous word. The same way a sentence may also have two or more meanings like "I saw a girl with a telescope" in this there are multiple meaning of the sentence, one is that the girl was carrying the telescope and another one is that the girl was seen through the telescope.

Hence, to avoid this we can use software to create a tool which may help us find the possible ambiguous words and correct them instantly. This may help the company to create more accurate requirements with minimum ambiguity in them.

II. LITERATURE REVIEW

As we all know that Ambiguity is the major problem during Requirement Elicitation. Hence, I am trying to create a concept with the help of which this problem could be solved. In this tool we need two software one for finding out all the possible Interpretations of a word and another for finding out the Part of Speech of the sentence provided by the Elicitor.

For finding out all possible interpretations we can use WordNet 2.1 Software developed at Princeton University Cognitive Science Lab by David Slomin and Randee Teng.

For finding the Part of Speech of a sentence we can use Stanford.NLP.POSTagger 3.5.2.1

We can merge both of these tools to create an ultimate tool which can be used for avoiding the ambiguity.

The working of this tool is very simple, the elicitor will type the information provided by the stakeholders each time we enter a point. The system will send each word of that sentence one by one to the WordNet tool which will find all the possible interpretations (under two categories only Noun and Verb) of that word and if the number of interpretations is more than one, the WordNet will return all the interpretation related to that work to our tool. Here at our tool the same point or sentence will be sent to POS Tagger tool to find out, to which part of speech the words in that sentence belong. And on the basis of the POS the interpretations of each word will be listed under a particular category (Noun and Verb) with that word. Now the Elicitor can choose the particular word which he finds suitable for the sentence after discussing about it with the stakeholders.

After that if we find that we are satisfied with our work we can further save it in the form of a DOC file.

A tool named Systemized Requirements Engineering Environment (SREE) [2] was created to avoid the ambiguity in the Requirement specification. In this tool the whole document regarding the requirement was loaded and the tool identifies most of the ambiguity in the requirement specification. SREE searches for instances of potential ambiguity in its input, concentrating on achieving a 100% recall rate and a precision rate of as close as possible to 100%. The potential ambiguities that SREE searches or are the ones whose indicators are listed in SREE's ambiguity indicator corpus (AIC).

In this tool they are using Wordnet 3.0 to find out the ambiguous. This tool also uses the part-of-speech tagger. For documentation the elicitor must keep in mind the various rules designed for elicitation to bring out the best result.

The result generated by SREE is the indication of almost all the possible ambiguities present in the document.

Another tool named Ambiguity Detector [3] was developed. The idea was very simple which does not require a large amount of rules and techniques. The tool created has three major parts Corpus, Part-of-Speech tagger and an algorithm to detect the possible ambiguities in the document. Here, Corpus is the collection of ambiguous words and phrases, Part-of-Speech tagger is used to identify that what part of speech that word belongs to and the algorithm is helping in identifying the ambiguity with the help of Corpus and Part-of-Speech tagger. The Part-of-Speech tagger helps in identifying the lexical/syntactic/ syntax type of ambiguities. There are no specific rules

for gathering the requirement and the result provided by the tool is with proper indication of ambiguous words. The found ambiguous words are colour coded and their count is also displayed.

And then Henrik Leopold, Fabian Pittke, and Jan Mendling [4] worked together and tried to create a system which can automatically detect and resolve the Lexical Ambiguity in process models and their main target is the Homonyms and Synonyms. For automatic analysis, lexical databases such as WordNet [5], [6] are available, which capture various semantic relationships in a structured way. A similar resource is the BabelNet database [7], which combines WordNet senses with the web encyclopedia Wikipedia and allows also multilingual word sense retrieval and disambiguation.

There is a very simple logic to detect Homonyms and Synonyms. For identification of Homonyms, every word having more than one sense as per the Wordnet software is considered to be a homonym. Then the correct sense is selected and used in the document. To identify Synonyms, the technique is to consider pairs of words that have at least one meaning in common. Those two words can be synonymous only if their context is approximately identical.

III. PROPOSED METHODOLOGY

A space will be provided in the tool where we can type each information and check for ambiguity after each point, remember we are avoiding ambiguity instantly instead of wait for the work of elicitation to finish

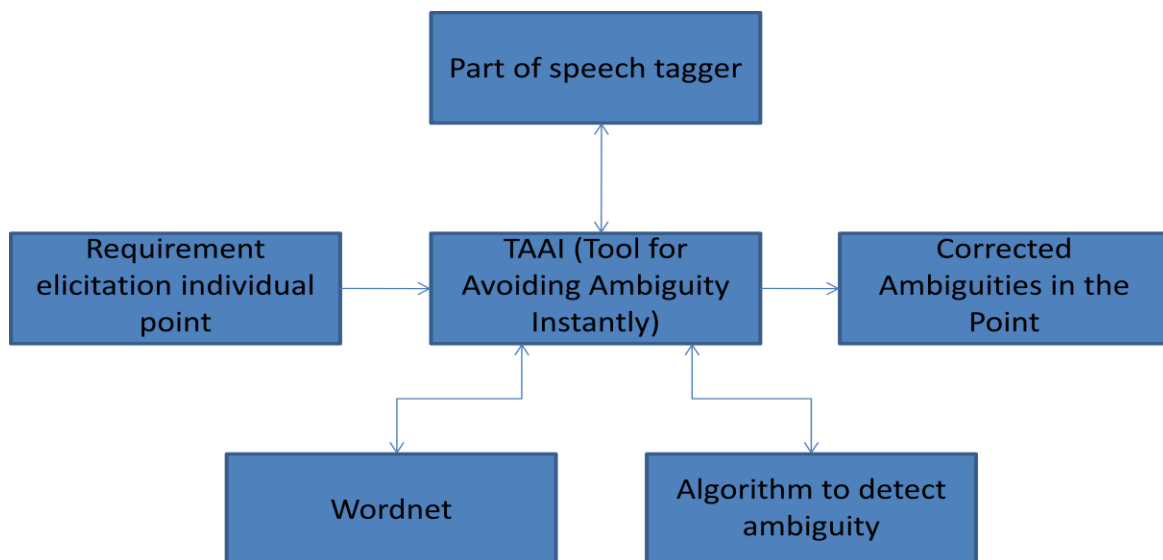


Figure 1 Architecture of TAAI (Tool for Avoiding Ambiguity Instantly)

- Enter the point in a space provided and then click a button to check ambiguity.
- Then each word in that sentence will be used to find the number of interpretations of a particular word in noun and verb form using WordNet 2.1 which will work with this tool.
- If the particular word's interpretation is more than one then that word will be listed below with all the possible interpretations.
- The interpretations of that word will be in a particular order which will be based on P.O.S. (Part of Speech) of that word. The ordering will be based on likelihood of that word.
- The Elicitor can select the most appropriate word after consulting it with the stakeholder.
- After selecting the correct interpretation of the words in a point and the elicitor presses enter all the selected interpretations will be displayed with their words in a bracket and if the elicitor finds it correct he can add the data to the DOC file.

Part of this system

- User Interface- In the interface there is a textbox to type the points provided by the stakeholders. And right below it is a button to check for the ambiguous words. All the ambiguous words will be displayed below with the senses in the dropdown list. Those senses will also be arranged on the basis of that word being noun and verb. After selecting the appropriate sense click on the submit button to see the result. The result will be a sentence with selected sense written in brackets with the ambiguous words. In this way we will show the correct sense of the sentence without changing the sentence. If the elicitor and the stakeholder are satisfied with the result that point will be then included in the sentence.
- Software Required- Here, we can use WordNet 2.1[9] to find out the sense of each word in that sentence on the basis of noun and verb. But before doing that we will try to find out the part of speech of each word using Stanford.NLP.POSTagger 3.5.2.1.[10] This will help in setting the priority of the senses extracted from the WordNet 2.1, like if the word is used as a verb in that sentence then verb related senses will be displayed first and then the noun based one. All the senses will be displayed in a dropdown list and from those we can select the appropriate one to be displayed in the result.
- WordNet 2.1- WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations.

The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.[11]

- POSTagger- A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'.[12], [13]
- TAAI (Tool for Avoiding Ambiguity Instantly)- This tool is combined with two different software one to find the sense of the given word and another to find the part of speech using their APIs. The software which provides the senses has every possible sense stored in it which on search will be sent to the tool. The software which we are using to find the part of speech is capable of finding the possible correct part of speech in a sentence and after processing the result will be sent to tool. Then this tool will sent the ambiguous word with the part of speech to the software to search for the senses on the basis of its part of speech. Then the result will be displayed in TAAI in a dropdown list.

IV.CONCLUSION

We know that Ambiguity is the major problem in requirement elicitation but still it is a problem which does have a solution. The above discussed method if developed can solve this problem because the above mentioned technologies are available and can be brought together to use it in salving our problem.

Using this method we can convert ambiguous sentences into lucid sentence. This technique may reduce the misinterpretations due to which the whole project suffers. This tool's result will be displayed old sentence with the correct meaning of the ambiguous words in the braces. In this way the stakeholder will not be able to accuse anyone for fudging the results.

So, in the future this tool is going to be very important in software/web development process saving huge amount of money and time. Not only this, it can be developed for various languages[15].

REFERENCES

- [1] Husain M. Shahid, M. R. Beg, "Word sense ambiguity: A survey" in International journal of computer and information technology (IJCIT), 2(6), PP. 1161-1168, 2013.
- [2] Sri Fatimah Tjong, "Avoiding Ambiguity in Requirements Specifications". Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy, February, 2008.
- [3] Neeraj Arya, Ayan Nigam, Deepika Jain, Bhawna Nigam, "Tool for Automatic Discovery of Ambiguity in Requirements", International Journal of Computer Science (IJCSI), 9(5), No 2, 2012

- [4] Fabian Pittke, Henrik Leopold, Jan Mendling, "Automatic Detection and Resolution of Lexical Ambiguity in Process Models". DOI 10.1109/TSE.2015.2396895, IEEE Transactions on Software Engineering, 2015.
- [5] G. A. Miller , "WordNet: a Lexical Database for English", Communications of the ACM, 38(11), PP. 39-41, 1995.
- [6] G. Miller and C. Fellbaum, "WordNet: An Electronic Lexical Database", Cambridge, MA: MIT Press, 1998.
- [7] R. Navigli and S. P. Ponzetto, "Multilingual. WSD with just a few lines of code: the BabelNet API" in ACL System Demonstrations, pp. 67-72, 2012.
- [8] Ben Kovitz, "Ambiguity and What to Do about It" Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02) 1090-705X/02, 2002.
- [9] Wordnet Software can be downloaded from this link <https://wordnet.princeton.edu/wordnet/download/current-version/>
- [10] Part-of-Speech tagger the source code can be found at <https://sergey-tihon.github.io/Stanford.NLP.NET/StanfordPOSTagger.html>
- [11] <https://wordnet.princeton.edu/>
- [12] Kristina Toutanova, Christopher D. Manning, "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger" in Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), PP. 63-70, 2000 .
- [13] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer, "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network" in *Proceedings of HLT-NAACL 2003*, PP. 252-259, 2003.
- [14] Husain M. Shahid, M. R. Beg, "Advances in ambiguity less NL SRS: A review" in the proceeding of IEEE International conference on engineering and technology (ICETECH), PP. 221-225, 2015.
- [15] Husain M. Shahid, Preeti, Yadav, "Study Hindi word sense disambiguation based on Hindi WordNet" in International journal for research in applied science and engineering technology, 2(5), PP. 390-395, 2014.