Understanding Ambiguities in Natural Language Requirements: A Systematic Review[1]

Srinivas Babu Ratnam

Cloud Data Architect

Date of Receiving:14 May 2023;
Date of Acceptance:23 June 2023;
Date of Publication: 03 July 2023

ABSTRACT

Natural Language Processing (NLP) is an application of artificial intelligence that focuses on understanding human language. Ambiguity analysis, one of the five segments of NLP, deals with context and identifying instances where a statement is not specific enough. Machine learning techniques can be applied to classify ambiguities, and with the availability of digital data, these techniques can be highly effective. In this paper, a systematic review is presented to explore the ambiguities arise in NLP requirements. The paper is mainly focused on language ambiguities such as semantic, anaphoric, lexical and pragmatic and also highlighted their issues. Then paper presented a meta-analysis of recent contributions and approaches to detect and resolve such ambiguities. After analysis of several methods some of the most contributing methods to resolve ambiguities are such as controlled language, knowledge based, controlled natural languages, machine learning. Therefore, this paper concludes that hybridization of such techniques will work more efficiently.

Keywords: *Natural Language Processing; Machine Learning; Semantic; Language; Ambiguities*

INTRODUCTION

The development of intelligent artificial agents has made crucial subfields of AI, such as machine learning and natural language processing, indispensable. Speech recognition, NLU, and NLG are just a few of the processes that NLP employs to produce a naturalistic answer. NLP has advanced significantly and is now recognized as the primary technology powering popular virtual assistants. Although NLP technology has advanced, users still frequently encounter ambiguity errors when retrieving answers [1]. Natural question language (NQL) processing, document preparation, and answer processing are the three main steps of the Question Answering (QA) system, which tries to address ambiguous mistakes in natural language processing. The process of detecting the right meaning of a word based on its usage, such as the word "bank," which can be used to refer to a building, a financial institution, or a river bank, is known as word sense disambiguation and is a crucial component of NLP. Researchers have been studying word sense disambiguation for a long time in order to improve NLP technologies [2].The problem of absolute disambiguation in natural language processing is considered a challenging problem in strong AI [3]. Ambiguity arises in NLP when a sentence can be interpreted in more than one way. Predicate logic has been used to formalize meaning, but non-declarative statements pose a challenge [4]. Semantic networks and frames have been developed to address this challenge, but context plays a significant role in interpreting meaning. Pragmatics, which studies how context affects meaning, is essential for understanding why an utterance is made, but it is not as well researched as syntax and semantics [5]. Therefore, this paper is dedicated to review existence of different ambiguities that arise in natural language processing. The paper presented a systematic review on different tools and techniques to resolve language ambiguity in NLP requirements.

NATURAL LANGUAGE PROCESSING REQUIREMENTS AND RELEVANT AMBIGUITIES

NLP is a branch of AI centered on creating models and algorithms that can analyze, comprehend, and produce human language. Due to the expansion of social media platforms, the growth of text data on the internet, and the development of deep learning techniques, NLP has become much more popular in recent years. NLP has several uses in a variety of industries, including chatbots, voice-based assistants, spam detection, sentiment evaluation, extraction of data, and question-answering systems [6]. The basic objective of NLP is to make it possible for machines to comprehend and

[1] *How to cite the article:* Ratnam S.B. (July 2023); Understanding Ambiguities in Natural Language Requirements: A Systematic Review, *International Journal of Law, Management and Social Science*, Vol 7, Issue 3, 1-8

produce human language similarly to how humans do. This is a difficult undertaking because human language is confusing, complicated, and context-dependent by nature. For instance, the term "bank" can be used to describe both a financial organization and a riverbank. In order to correctly interpret a word's meaning, NLP algorithms must be able to understand the context in which it is used. Uncertainty is one of the biggest problems that NLP systems have to deal with. When a term or phrase might signify different things based on the context in which it is used, this is called ambiguity. As an illustration, the phrase "I saw her duck" can also mean "I observed her bird" or "I watched her lower her head." To address this issue, NLP systems employ a variety of strategies, including analysis of context, machine learning, and deep learning, to determine the text's intended meaning. NLP is a rapidly expanding field with many uses across several industries. Despite the difficulties created by the complexity and ambiguity of human language, the success of NLP systems rests on their capacity to effectively assess and convey the intended meaning of the text. Some of the requirements of NLP are presented below:

- Machine Translation: a tool for translating text from one language to another, which can be challenging due to differences in syntax and structure. Google Translate uses statistical engines to translate words between languages, while recent systems are based on deep learning and neural networks [7].
- Text Categorization: a system for allocating information to predetermined categories or indexes, such as official papers, market data, or complaint requests. Spam filters are an example of text categorization tool, but they can suffer from false positives or negatives [8].
- Spam filtering: a method that uses ML and text categorization to separate spam emails from real communications [9].
- Finding important terms in textual data, such as people, places, events, dates, times, and prices, is a technique known as information extraction. This can be used to generate databases, summarize information, and classify text into predetermined categories [10].
- Summarization: a technique for condensing information while keeping its significance, which enables us to understand more in-depth emotional meanings and pertinent details for a large amount of information. It can be utilized as a marketing tool, for example, to examine the overall sentiment on social media [11].
- Dialogue System: a conversation system assistance tool that demands in-depth context awareness. Platforms like Google Assistant, Cortana for Windows, Siri for Apple, Alexa for Amazon, and many more that enable robots to converse with individuals using natural language have been developed as a result.
- Medicine: NLP is used in the medical industry to detect potential side effects of any medication, summarize information of any signs or symptoms, drug dose, and response data, and assist clinicians in extracting and highlighting data items[12].

Natural Language Processing (NLP) faces processing challenges due to the ambiguities present in natural language. The ambiguities can arise due to various reasons, such as the use of synonyms, homonyms, polysemous words, and idioms. Ambiguity in the software development process can cause errors and inefficiencies. Ambiguity can arise due to missing information and communication errors. The passage categorizes ambiguity into two types of errors: language and requirement, as presented in fig 1. For example, grammatical errors are a type of error that does not require domain knowledge, while errors related to a lack of information require domain knowledge. The passage also describes two types of ambiguity: linguistic ambiguity, between requirement-specific software ambiguity, which needs domain knowledge to discover, and general software ambiguity, which may be identified by any user who is familiar with the language. Most ambiguity-related mistakes in software development are caused by faulty requirements. Linguistic errors are relatively rare.

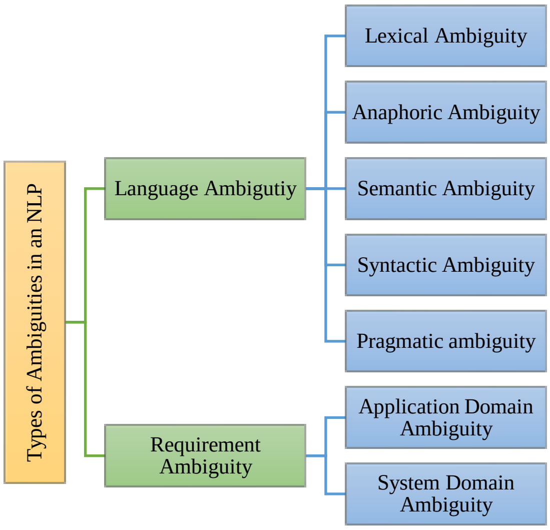

Fig. 1. Types of Ambiguities in an NLP

LANGUAGE AMBIGUITY IN NLP

In the context of artificial intelligence, developing natural language processing (NLP) software that can comprehend and respond to human language with accuracy is a serious issue. The purpose of human language is multilayered and includes pragmatic, semantic, and syntactic levels. For NLP systems to deliver precise and useful responses, they must be able to comprehend and interpret various levels of meaning. Although language is a natural means of communication, computers require structured data to process information accurately. However, using natural language in software specifications has several advantages, such as enhanced stakeholder dialogue and understanding. Natural language can introduce ambiguities that may cause misunderstandings and errors while developing software. Ambiguity detection in natural language requirements (NLRs) aims to identify and eliminate such ambiguities to improve the quality and correctness of requirements. Lack of information, uneven language use, or unstated presumptions can all lead to ambiguities. It is difficult and requires understanding of the domain being created and the environment in which the requirements are stated to identify ambiguity in NLRs. Failure to detect and resolve ambiguity in NLRs can result in delays, expensive errors, and a system that fails to meet user requirements[7]. There are different types of language ambiguities that can occur in natural language processing (NLP). These include lexical ambiguity, which happens when a single word has several meanings, syntactic ambiguity, which develops when a statement's structure allows for multiple interpretations, semantic ambiguity, which happens when a word or phrase has several meanings, and others, anaphoric ambiguity, which occurs when ambiguity arises from the use of anaphora, and pragmatic ambiguity, which arises due to the context in which a statement is made and can have multiple interpretations. The process of ambiguity detection involves several steps (fig 2): first, potential ambiguities in the natural language requirements are identified, either through NLP tools or expert review. To resolve ambiguities in software development, the context of the ambiguous statements is analyzed, and the intended meaning is understood. Then, the requirements are clarified by adding information or rephrasing them. The resolved ambiguities are then validated to ensure they meet stakeholders' needs, either through user testing or formal review by experts.

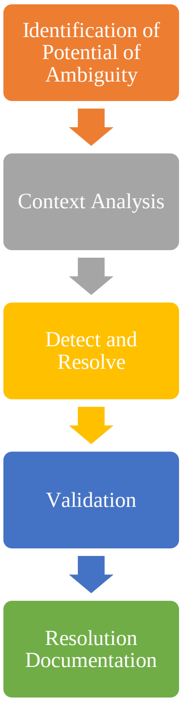

Fig. 2. Steps to Detect the Ambiguities

DETECTION TECHNIQUES FOR LANGUAGE AMBIGUITY IN NLP

Ambiguity in Natural Language Processing (NLP) is one of the major difficulties that NLP systems must overcome. When a word or phrase can signify different things according to the context in which it is used, this is referred to as language ambiguity. Part-of-Speech (POS) tagging, that provides every single word in a sentence a distinct grammatical subject matter, is one of the frequently utilized detection approaches. By categorizing words according to their functions in a sentence and homonyms or polysemous words according to their part of speech, POS tagging helps to clarify the literal meaning of words. Word Sense Disambiguation (WSD), which entails determining the appropriate sense of a word in a given context, is a different approach for recognizing linguistic ambiguity. To determine a word's intended meaning, WSD algorithms examine the context whereby it is used, including the words that come before and after it. Another method for detecting and resolving ambiguity in language used in NLP is contextual ambiguity resolution. In order to clarify a word or phrase's meaning using this technique, the larger context in which it is used must be examined. Contextual Ambiguity Resolution combines statistical models and machine learning algorithms to deduce the intended meaning of a word or phrase while taking into consideration the entirety of the document or discussion.

Yadav et al. [11] conduct a comprehensive review of various disambiguation tools and analyze existing research work. They note that some of these tools are still under development and may be capable of eliminating ambiguities in the future. Ferrari et al. [12][18] To identify pragmatic ambiguity in natural language needs, a proposed graph-based modeling technique utilizing a shortest-path search algorithm and a web-search engine was put forth. Zait et al. [13] offer a method for ambiguity detection and settlement in conversational requirements utilizing semantic web and NLP approaches. Kato et al. [14] to detect unclear language in draft procurement requirements and to reach a maximum F value of 0.19, develop a knowledge dictionary. Ashfaq et al. [15] aim to capture stakeholder's requirements using SBVR-based CNL and prepare a semantically consistent SRS document. Osama et al. [16] With four filtering pipelines, we suggest an effective and flexible automatic syntactic ambiguity detection method that, on average, achieves 65% precision and 99% recall. Sharma et al. [17] consider the task of classifying ambiguous requirements statements with pronominal anaphora ambiguity and achieve a recall of 95% with a Bayesian network classification algorithm. Mishra et al. [19] apply word embeddings to detect domain-specific ambiguities in frequently used computer science words. Ferrari et al. [20] use SREE, a tool that searches for typically ambiguous terms, to detect defects in industrial requirements and show that it may complement pattern-based techniques.Manam et al. [21] introduce TaskLint, a system to detect problems with task instructions using NLP tools. Sütçü et al. [22] propose using sentiment analysis to analyze movie reviews on beyazperde.com. Ferrari et al. [23] present a method for using processing of natural languages to find ambiguous terms in various domains. Ceccato et al. [24] introduce a prototype tool for identifying and measuring ambiguity in natural language text. Sallis et al. [25] provide a selection of recently published papers on ambiguity in natural language. Yuwan et al. [26] discuss ambiguity in natural language comprehension and compare human-computer language grammar ambiguity. Dhopavkar et al. [27] Present their work on Marathi text syntactic annotation using a rule-based method, and talk about how a morphological analyzer might be used to detect gender during syntactic analysis. Shrestha et al. [28] investigate the impact of NLP resources like morphology analyzer and machine-readable dictionary in ambiguity resolution. Ezzini, et al.[29] addressed the ambiguity detection and anaphora interpretation to solve the anaphoric type language ambiguity. Their method had an average success rate of 98% for anaphora interpretation and an average precision of 60% and recall of 100% for anaphoric ambiguity detection. However, SpanBERT was found to be the most accurate solution for anaphora interpretation. Ezzini et al.[30] developed an automated method for handling requirements ambiguity that makes advantage of natural language processing. Our method produces automatic interpretations with an average accuracy of 85%. Our method, which makes use of domain-specific corpora, has 33% better ambiguity detection accuracy and 16% better interpretation accuracy when compared to foundations that use generic corpora. Sun et al. [31] introduce a new method of ambiguity discovery, while Hou et al. [32] a new unsupervised approach for Chinese WSD should be proposed. Roopa et al. [33] describe a supervised neural network model for sense detection, and Kaddoura et al. [34] present a survey of research works on Arabic word sense disambiguation.Saxena et al. [35] assemble SMT systems for five translation jobs that demonstrate gains over the baseline model in terms of the evaluation metrics BLEU and METEOR. Below table 1 presents the summary of applications used to handle ambiguities in NLP requirements. Below in fig 4, summery of the presented works are represented graphically and it was observed that controlled natural language outperforms the best.

TABLE I. PERFORMANCE EVALUATION OF TECHNIQUES TO MITIGATE AMBIGUITIES IN NLP REQUIREMENTS

Ref	Methodology	Technologies	Ambiguity	Results
[11]	Managed Language	POS Tagging	Lexical	Recall 80.12% and Precision 85.76%
[11]	Ontology-based knowledge	Stanford	Lexical	Recall 92.85% Precision 92.85%
[11]	Knowledge based & Ontology	WordNet	Lexical	Precision 83.4%
[12]	ML	WordNet	Lexical	-
[13]	ML	Stanford	Semantic	-
[14]	Knowledge Dictionary	-	Pragmatic	Recall 99%
[15]	Controlled Natural Languages	SBVR vocabulary	Semantic	Recall 97% precision 96%
[16]	Filtering pipelines	Stanford	Syntactic	Recall 99% precision 65%
[17]	Bayesian network	-	Pronominal Anaphora	Recall 95%
[18]	Collective Intelligence	-	Pragmatic	-
[20]	ML	Word embedding	-	Precision 83% and recall 85%
[22]	Naive Bayes	Turkish Movie Reviews Emotion Dictionary	Pragmatic	Sensitivity 85%
[23]	Language models	Word embedding	cross-domain	-
[24]	-	WordNet	lexical	-
[25]	Neural Network	-	Linguistic	-
[26]	Statistical rules	-	Grammar	-
[27]	Ruled-based approach	-	Grammar	-
[28]	morphology analyzer	machine readable dictionary	Grammar	Accuracy 80%
[29]	BERT	-	Anaphoric	Precision 60%

[30]	Domain-Specific Corpora	-	Domain	Precision 80% recall 89%

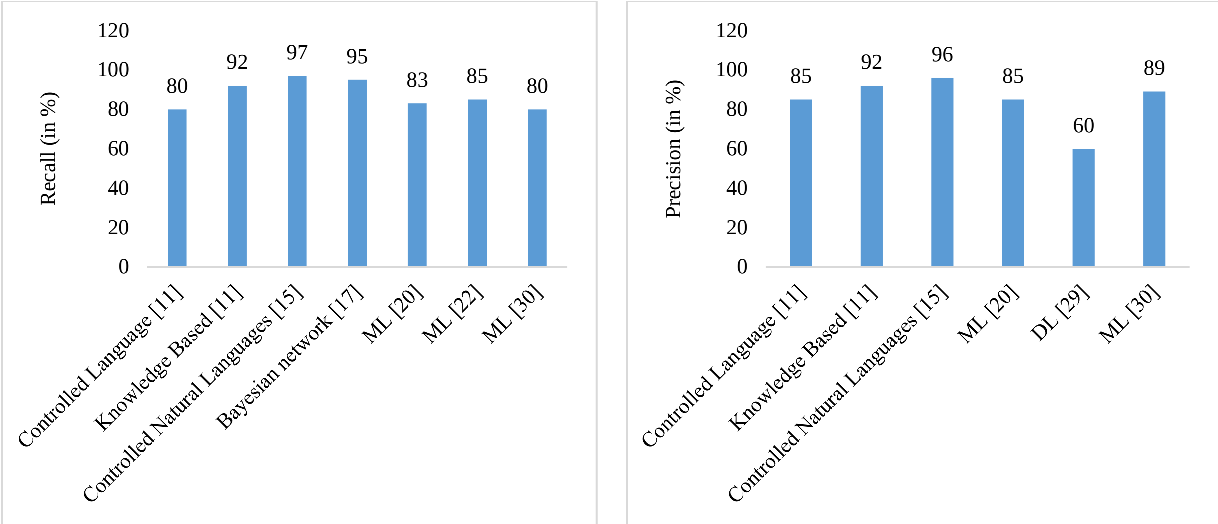

Fig. 3. Comparative State-of-Art Techniques to Resolve Ambiguity in NLP

CURRENT RESEARCH CHALLENGES

NLP (natural language processing) faces various challenges such as contextual words and phrases, synonym processing, linguistic complexity levels, homophones, irony, sarcasm, and ambiguous statements. Dealing with language specific to certain regions, informal phrases, idioms, and misspelled or misused words are also difficult. Although NLP models for commonly spoken languages are developing, rather than a specialized understanding of a specific language and technology, models for all individuals are still required. The automation of ambiguity detection is proposed as a solution, with a tool that provides trustworthy and accurate identification of ambiguities while explaining their causes. This tool can also be used for analyst training, time and money savings, and improved quality of industrial requirements engineering. Developing a pragmatic ambiguity detector faces challenges such as natural language understanding, domain-specific knowledge, inconsistency, annotation, evaluation, scalability, adaptability, and false positives. To overcome these challenges, detector should be well-trained, use NLP techniques, and have domain-specific expertise, evaluated and improved over time. It should also be scalable, adaptable, and minimize false positives.

CONCLUSION

This paper emphasizes the importance of recognizing pragmatic ambiguities in natural language requirements to ensure the overall quality and accuracy of software requirements. The paper compares various approaches for identifying and resolving ambiguity in natural language requirements and makes the case that improved approaches will be produced by further developments in NLP, ML, and AI. The purpose of the study is to determine how ambiguous common computer science phrases are when employed in various settings. The findings show that the ambiguity issues are accurately identified and resolved by the tool that was developed. The algorithm, however, only recognizes one sort of ambiguity and ignores other types of ambiguity, such as semantic, syntactic, lexical, or pragmatic ambiguity. According to the study's findings, as the field advances, ambiguity detection will play a bigger role in conversational systems, automated language translation, and natural language comprehension.

REFERENCES

1. Ferrari, G. Lipari, S. Gnesi and G. O. Spagnolo, "Pragmatic ambiguity detection in natural language requirements," 2014 IEEE 1st International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), Karlskrona, Sweden, 2014, pp. 1-8, doi: 10.1109/AIRE.2014.6894849.
2. Ortega-Martín, Miguel, Óscar García-Sierra, Alfonso Ardoiz, Jorge Álvarez, Juan Carlos Armenteros, and Adrián Alonso. "Linguistic ambiguity analysis in ChatGPT." arXiv preprint arXiv:2302.06426 (2023).

3. M. Q. Riaz, W. H. Butt and S. Rehman, "Automatic Detection of Ambiguous Software Requirements: An Insight," 2019 5th International Conference on Information Management (ICIM), Cambridge, UK, 2019, pp. 1-6, doi: 10.1109/INFOMAN.2019.8714682.
4. Sabriye, A. O. J. a., & Zainon, W. M. N. W. (2017). A framework for detecting ambiguity in software requirement specification. Paper presented at the 2017 8th International Conference on Information Technology (ICIT).
5. Ali, S. W., Ahmed, Q. A., & Shafi, I. (2018). Process to enhance the quality of software requirement specification document. Paper presented at the 2018 International Conference on Engineering and Emerging Technologies (ICEET).
6. Yue Kang, Zhao Cai, Chee-Wee Tan, Qian Huang & Hefu Liu (2020) Natural language processing (NLP) in management research: A literature review, Journal of Management Analytics, 7:2, 139-172, DOI: 10.1080/23270012.2020.1756939
7. Dhar, A., Mukherjee, H., Dash, N.S. et al. Text categorization: past and present. Artif Intell Rev 54, 3007–3054 (2021). https://doi.org/10.1007/s10462-020-09919-1
8. P. Garg and N. Girdhar, "A Systematic Review on Spam Filtering Techniques based on Natural Language Processing Framework," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, pp. 30-35, doi: 10.1109/Confluence51648.2021.9377042.
9. S. Alves, J. Costa and J. Bernardino, "Information Extraction Applications for Clinical Trials: A Survey," 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 2019, pp. 1-6, doi: 10.23919/CISTI.2019.8760639.
10. Abualigah, L., Bashabsheh, M.Q., Alabool, H., Shehab, M. (2020). Text Summarization: A Brief Review. In: Abd Elaziz, M., Al-qaness, M., Ewees, A., Dahou, A. (eds) Recent Advances in NLP: The Case of Arabic Language. Studies in Computational Intelligence, vol 874. Springer, Cham. https://doi.org/10.1007/978-3-030-34614-0_1.
11. A.Yadav, A. Patel, and M. Shah, “A comprehensive review on resolving ambiguities in natural language processing,” AI Open, vol. 2, pp. 85–92, 2021, doi: https://doi.org/10.1016/j.aiopen.2021.05.001.
12. Ferrari, G. Lipari, S. Gnesi, and G. O. Spagnolo, “Pragmatic ambiguity detection in natural language requirements,” in 2014 IEEE 1st International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), 2014, pp. 1–8. doi: 10.1109/AIRE.2014.6894849.
13. F. Zait and N. Zarour, “Addressing Lexical and Semantic Ambiguity in Natural Language Requirements,” in 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), 2018, pp. 1–7. doi: 10.1109/ISIICT.2018.8613726.
14. T. KATO and K. TSUDA, “A Method of Ambiguity Detection in Requirement Specifications by Using a Knowledge Dictionary,” Procedia Comput. Sci., vol. 207, pp. 1482–1489, 2022, doi: https://doi.org/10.1016/j.procs.2022.09.205.
15. F. Ashfaq and I. S. Bajwa, “Natural language ambiguity resolution by intelligent semantic annotation of software requirements,” Autom. Softw. Eng., vol. 28, no. 2, p. 13, 2021, doi: 10.1007/s10515-021-00291-0.
16. M. Osama, A. Zaki-Ismail, M. Abdelrazek, J. Grundy, and A. Ibrahim, “Score-Based Automatic Detection and Resolution of Syntactic Ambiguity in Natural Language Requirements,” in 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2020, pp. 651–661. doi: 10.1109/ICSME46990.2020.00067.
17. R. Sharma, N. Sharma, and K. K. Biswas, “Machine Learning for Detecting Pronominal Anaphora Ambiguity in NL Requirements,” in 2016 4th Intl Conf on Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD), 2016, pp. 177–182. doi: 10.1109/ACIT-CSII-BCD.2016.043
18. Ferrari and S. Gnesi, “Using collective intelligence to detect pragmatic ambiguities,” in 2012 20th IEEE International Requirements Engineering Conference (RE), 2012, pp. 191–200. doi: 10.1109/RE.2012.6345803.
19. S. Mishra and A. Sharma, “On the Use of Word Embeddings for Identifying Domain Specific Ambiguities in Requirements,” in 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), 2019, pp. 234–240. doi: 10.1109/REW.2019.00048.
20. Ferrari et al., “Detecting requirements defects with NLP patterns: an industrial experience in the railway domain,” Empir. Softw. Eng., vol. 23, no. 6, pp. 3684–3733, 2018, doi: 10.1007/s10664-018-9596-7.
21. V. K. C. Manam, J. D. Thomas, and A. J. Quinn, “TaskLint: Automated Detection of Ambiguities in Task Instructions,” Proc. AAAI Conf. Hum. Comput. Crowdsourcing, vol. 10, no. 1, pp. 160–172, 2022, doi: 10.1609/hcomp.v10i1.21996.

22. S. Sutcu and C. Aytekin, "AN EXAMPLE OF PRAGMATIC ANALYSIS IN NATURAL LANGUAGE PROCESSING: SENTIMENTAL ANALYSIS OF MOVIE REVIEWS," in TURKISH ONLINE JOURNAL OF DESIGN ART COMMUNICATION, 2019, vol. CTC, pp. 61–74. doi: 10.7456/ctc_2019_05.
23. Ferrari, A. Esuli, and S. Gnesi, "Identification of Cross-Domain Ambiguity with Language Models," in 2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), 2018, pp. 31–38. doi: 10.1109/AIRE.2018.00011.
24. M. Ceccato, N. Kiyavitskaya, N. Zeni, L. Mich, and D. Berry, "Ambiguity Identification and Measurement in Natural Language Texts," 2004.
25. P. Sallis, "Neural networks and ambiguity in natural language," in 1993 First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, Nov. 1993, p. 102,103. doi: 10.1109/ANNES.1993.323071.
26. G. Yuwan, F. Feng, Y. Ruimin, and S. Yuqiang, "The Comparative Research of the Ambiguity of Comprehension of Human-Machine Language," in 2009 First International Workshop on Education Technology and Computer Science, 2009, vol. 3, pp. 363–366. doi: 10.1109/ETCS.2009.608.
27. G. Dhopavkar and M. Kshirsagar, "Syntactic analyzer using morphological process for a given text in natural language for Sense Disambiguation," in 2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), 2014, pp. 911–914. doi: 10.1109/CONFLUENCE.2014.6949378.
28. N. Shrestha, P. Hall, and S. Bista, "Resources for Nepali Word Sense Disambiguation," 2008, doi: 10.1109/NLPKE.2008.4906758.
29. S. Ezzini, S. Abualhaija, C. Arora, and M. Sabetzadeh, "Automated Handling of Anaphoric Ambiguity in Requirements: A Multi-Solution Study," in Proceedings of the 44th International Conference on Software Engineering, 2022, pp. 187–199. doi: 10.1145/3510003.3510157.
30. S. Ezzini, S. Abualhaija, C. Arora, M. Sabetzadeh, and L. C. Briand, "Using Domain-Specific Corpora for Improved Handling of Ambiguity in Requirements," in 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 1485–1497. doi: 10.1109/ICSE43902.2021.00133.
31. T. Sun, Y. Liu, L. Yang, Z. Li, and Z. Liu, "An Ambiguity Discovery Algorithm on Chinese Word Segmentation Based Dictionary," in 2009 Second Pacific-Asia Conference on Web Mining and Web-based Application, 2009, pp. 39–42. doi: 10.1109/WMWA.2009.18.
32. Hou, F. Qi, Y. Zang, X. Zhang, Z. Liu, and M. Sun, "Try to Substitute: An Unsupervised Chinese Word Sense Disambiguation Method Based on HowNet," COLING 2020 - 28th Int. Conf. Comput. Linguist. Proc. Conf., pp. 1752–1757, 2020, doi: 10.18653/v1/2020.coling-main.155.
33. H. R. Roopa and S. Panneer Arockiaraj, "The Role of Artificial Neural Network in Word Sense Disambiguation (WSD)---A Survey," in Rising Threats in Expert Applications and Solutions, 2022, pp. 221–227.
34. S. Kaddoura, R. Ahmed, and J. D., "A comprehensive review on Arabic word sense disambiguation for natural language processing applications," WIREs Data Min. Knowl. Discov., vol. 12, 2022, doi: 10.1002/widm.1447.
35. S. Saxena, U. Chaurasia, N. Bansal, and P. Daniel, "Improved Unsupervised Statistical Machine Translation via Unsupervised Word Sense Disambiguation for a Low-Resource and Indic Languages," IETE J. Res., vol. 0, no. 0, pp. 1–11, 2022, doi: 10.1080/03772063.2022.2098189.