Automatic Word Sense Disambiguation for Two Polysemous Words in a Single Line

Jaspreet Kaur*, Navjot Kaur
Department of Computer science & Engineering
Sri Guru Granth Sahib World University
Fatehgarh Sahib, Punjab, India.

Abstract- Word sense disambiguation (WSD) is the task of identifying or sensing the correct meaning of a word in a given context. Here the sense inventory comes from a dictionary or thesaurus to determine their different possibilities. In this paper, we present the methodology to develop an algorithm which is used to remove the ambiguity between two polysemous (ambiguous) words present within a single line. We enter lines which contain two similarly pronounced ambiguous words (Polysemous words). Our proposed algorithm tries to resolve this ambiguity by showing the meaning and other related detail corresponding to the ambiguous word. To measure the accuracy of our algorithm we perform testing on various lines. It was found in testing that the overall accuracy of the proposed algorithm is 65%.

Keywords-- ambiguity, word sense disambiguation, polysemous words, natural language processing, ambiguous words.

I.   Introduction

In natural languages, there are so many ambiguous words. There are near about 30% to 43% ambiguous words in English information and near about 42% ambiguous words in Chinese information. All natural languages contain words that can have different meanings of things in different contexts [7]. In English, for example, the word "Arm" can refer to a portable weapon or each of the two limbs of the human body from the shoulder to hand. Such words with multiple meanings are actually ambiguous, and the procedure of choosing which of their several meanings is purposive in a given context is known as Word Sense Disambiguation (WSD). English is the most commonly used language in the field of sciences. It is the widely spoken language and has frequently been referred to as a world language. Approximately 375 million people speak English as their first language. But the ambiguities present in this language create hindrance in usage of Information technology for native users [9]. Word sense disambiguation is the process of automatically figuring out the intended meaning of such a word when used in a sentence. Words can have different senses. Some words have multiple meanings. This is called polysemy or polysemous words [1]. The word **"Watch"**, for example, can mean

- to look at or
- a small time-piece usually worn on one's wrist.

Take another example of word **"Bear".** It can be:

- a large wild animal with thick fur,
- to put up with or to carry,
- to have a relation or connection or
- a dull, unpleasant, or difficult piece of work [6].

II.   Problem Defination

Ambiguity is often referred to as the major single challenge in NLP today. The central idea of this paper is to develop an algorithm which is used to remove the ambiguity between two polysemous (ambiguous) words which are occurred in a single line. Our proposed algorithm will show the meanings and other relevant information of those polysemous words which are present in the line e.g.

> **I have deposited money in a bank, which is situated at the bank side of river.**

- First of all the algorithm will split the input line into two parts. Various splitting criteria are defined in the algorithm.
- After that the algorithm will remove all the stop words from these two lines. Most Search Engines do not consider extremely common words in order to save disk space or to speed up search results. These filtered words are known as 'Stop Words'.
- Then it will pick the repeated word which occur in these two lines and consider that word as polysemous (ambiguous) word.
- After that the overlapping will perform and the algorithm shows the output detail corresponding to that word, like: the word, its meaning, synonyms etc.
- And finally the ambiguous words are replaced with their corresponding synonyms.

III. Proposed Algorithm

In this paper we develop an algorithm which is used to remove the ambiguity between two polysemous (ambiguous) words. The steps of algorithm are as follows:

Step 1: Select a line from database (D) or enter manually.

Step 2: Split the line where 'comma (,)', 'and', 'but', 'because' are found.

Step 3: For Line 1:

(i) Remove the stop words:-
   a) Compare string array with D of stop words one by one.
   b) If any match found, remove that word from the string.
   c) Similarly, remove all the stop words.
   d) Else load the remaining words into another string.

Step 4: For Line 2:

   Repeat step 3.

Step 5: Compare the Line 3 and Line 4 using string array.

(i) If repeated word found, load it into text box.

Step 6: For Line 3:

(i) Compare the string of remaining words with D of related words.
   a) Split Line 3 into string array.
   b) Compare the string array with related words column.
   c) If any match found, display all the data (information) corresponding to that word. i.e. word, meaning, related words, synonyms.
   d) After that, replace the repeated word with its corresponding synonym.

Step 7: For Line 4:

   Repeat step 6.

Step 8: Two output lines are shown in different text boxes.

IV. Implementation

In this paper, the polysemous words are considered which are occurred in a single line. To disambiguate these polysemous (ambiguous) words, we use overlapping based technique. Accuracy is an important issue to be considered. So, to measure the accuracy of our algorithm we perform experiments on number of lines. In developing this application MS Access is used as back end tool and VB .net is used as front end tool. Whenever the application is started, the window shown in Fig 1 will appear which contains the text area, where the user can enter the text to perform disambiguation. In this, we can choose a line from the database or enter a line manually.

Fig 2 shows that on the click on "Load Data" button, the various examples (lines) loaded in the database will be shown in text area. We can use the scroll bar for scrolling the text area to choose any line.
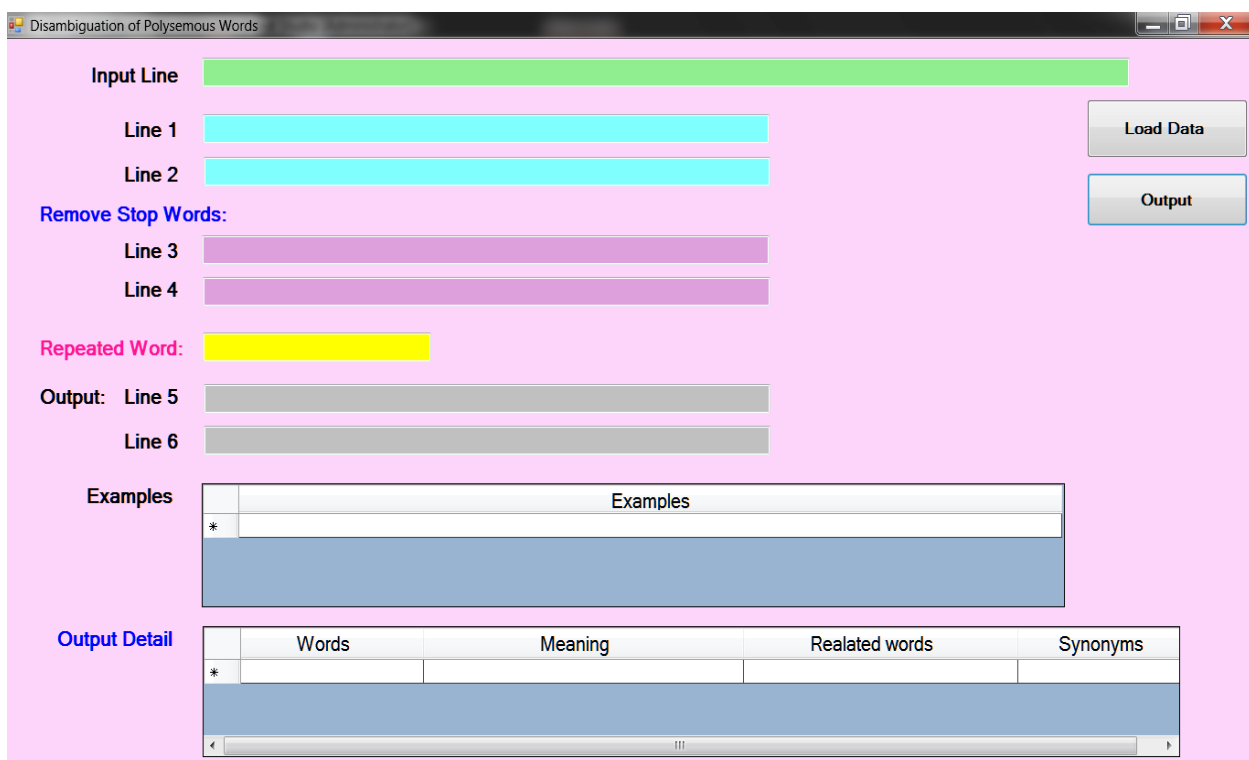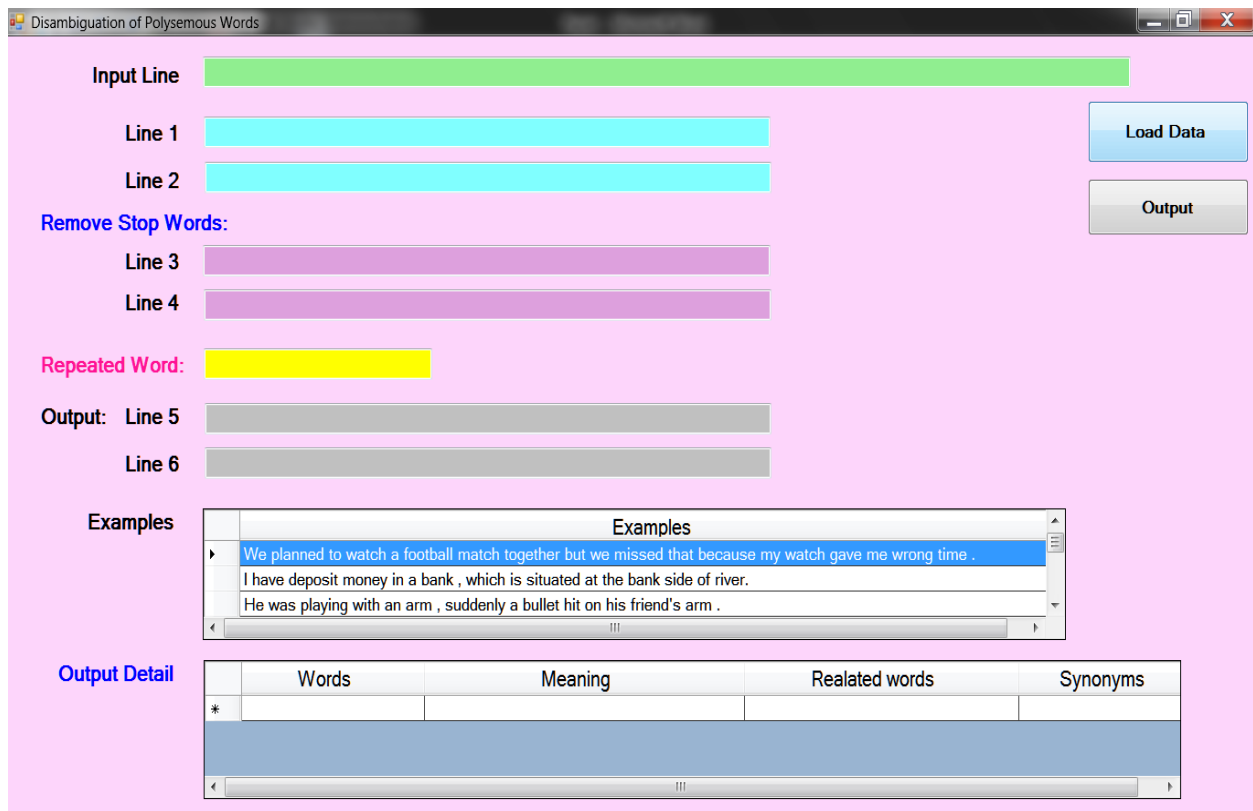

Fig 1: Snapshot of main screen.

Fig 2: Snapshot of working of Load Data button.

Fig 3 shows that after enter an input line, we click on "Output" button. After clicking the Output button, first of all the algorithm split the input line into two sections say Line 1 and Line 2 where the *comma (,), and, but* and *because* are used as splitting criteria. Then it will remove the stop words and the remaining words are shown in Line 3 and Line 4. From these two lines the repeated occurred word is chosen and treated as polysemous (ambiguous) word. Then the overlapping is performed with related words column. When the overlapping takes place, the algorithm show all the output details and replace the polysemous word with its corresponding synonym and give the result in Line 5 and Line 6.
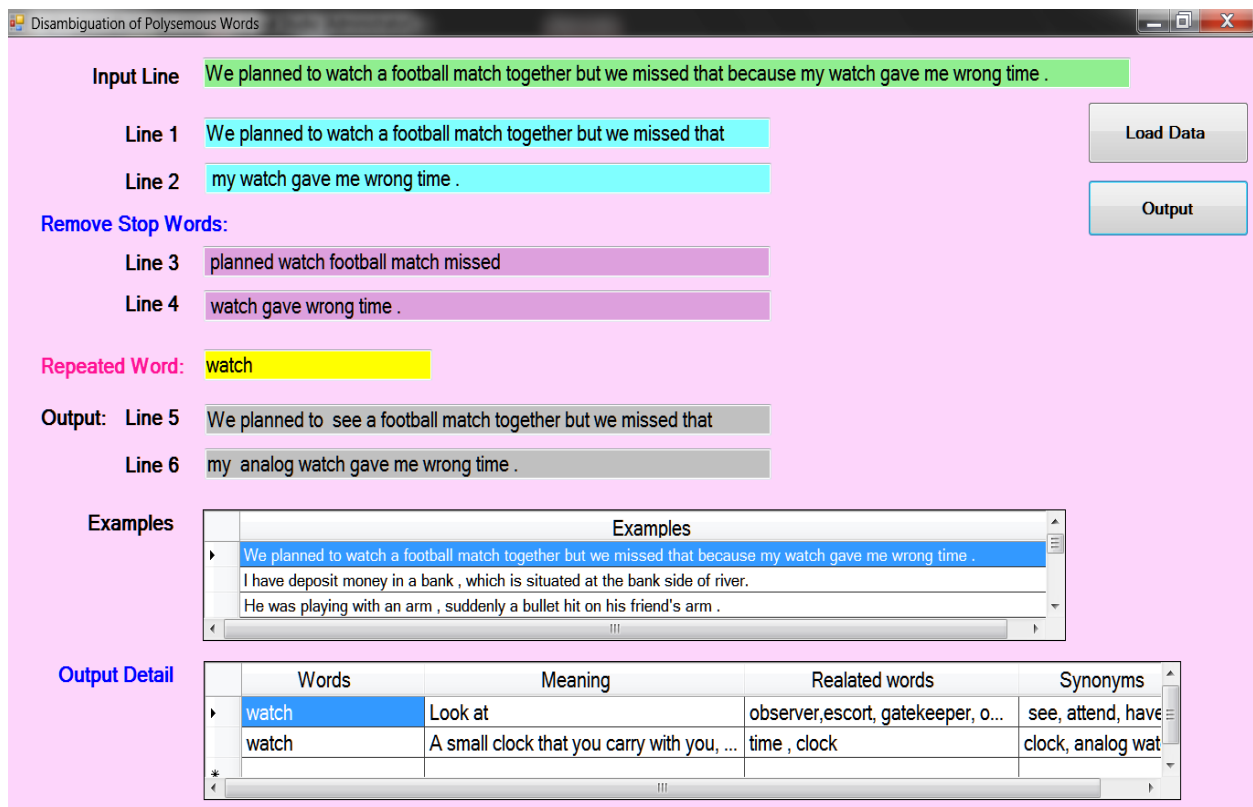
Fig 3: Snapshot of the Output Detail of the input line.

V.   Experiments And Results

The table shown below gives the results of output detail. It shows that whether the final results obtained are correct or not.

Table 1: Testing results of input lines.

Input Line and The Output Detail		Comments
Test Case 1	**I have deposited money in a bank[1], which is situated at the bank[2] side of river.**	
Output Detail	**Word: bank[1]** **Meaning**: a financial institution that people or businesses can keep their money in or borrow money from **Related words:** balance, bank balance, bank draft, money, banker's draft, banker's order, deposit, banking, bank rate, money, shares, cash, cheque, amount, rupees. **Synonyms:** investment firm, stock, repository, credit union, depository, central bank, saving bank, investment bank.	Correct
	Word: bank[2] **Meaning:** a raised area of land along the side of a river **Related words:** mound, heap, stack, mountain, pyramid, drift, river, canal hill side. **Synonyms:** bay, beach, coast, coastline, heap, hill side.	Correct
Test Case 2	**He was playing with an arm[1], suddenly a bullet hit on his friend's arm[2].**	
Output Detail	**Word: arm[1]** **Meaning:** a portable weapon from which a shot is discharged by gunpowder **Related words:** handgun, playing, pistol, revolver, rod, roscoe, sidearm, six-shooter, zip gun; self-loader, fieldpiece, rifle, shotgun, smoothbore, AK-47, assault rifle, assault weapon, play, shoot, training. **Synonyms:** firearm, pistol, knife, revolver, assault weapon, machine gun, machine pistol, submachine gun, Tommy gun, rifle.	Correct
	Word: arm[2] **Meaning:** one of the two long parts of your body with your hands at the end **Related words:** upper arm, limbs, upper arm, triceps, hit, blow, body, pain, fracture, muscles, hand, fingers, bone. **Synonyms:** armpit, hands, limbs, biceps, under arms, bow, elbow, forearm.	Correct
Test Case 3	**We planned to watch[1] a football match together but we missed that because my watch[2] gave me wrong time.**	
Output Detail	**Word: watch[1]** **Meaning:** Look at **Related words:** observer, escort, gatekeeper, observation, supervision, inspection, match, game, Tv, movie, film, play, supervise, look, theater, people, person. **Synonyms:** see, attend, have a look-see, inspect, keep an eye on, keep tabs on, look, observe, see, view, watch out, watch over.	Correct
	Word: watch[2] **Meaning:** A small clock that you carry with you, usually on your wrist **Related words:** time, clock, hours, minutes, wrist. **Synonyms:** clock, analog watch, chronometer, digital watch, pocket watch, stopwatch, ticker, timepiece, timer, wristwatch	Correct

Test Case 4	**Shren's mom was busy to prepare batter[1] and he batter[2] to slam the door with a crowbar.**	
Output Detail	**Word: batter[1]** **Meaning:** A mixture of flour, egg and water or milk used to making pancakes or coasting food **Related words:** baking powder, bicarbonate of soda, cream of tartar, egg white, gelatins, prepare, glaze, mix, oil, make. **Synonyms:** paste, mush, pancake, batter, puff batter, recipe, concoction, drought.	Correct
	Word: batter[2] **Meaning:** to strike repeatedly **Related words:** assail, assault, attack, hit, knock, punch, rush, slam, slap, smack, smash,spank, storm, whomp, fustigate, horsewhip, leather, pistol-whip, rawhide. **Synonyms:** hit, smack, bludgeon, thump, beat, beat up, stun, thrash, knock around, punch, thump, knock, bang.	Correct
Test Case 5	**She was standing in the shade[1] of sun, so the shade[2] of her dress was looking dull.**	
Output Detail:	**Word: shade[1]** **Meaning:** **Related words:** **Synonyms:**	No Match Found
	Word: shade[2] **Meaning:** **Related words:** **Synonyms:**	No match Found
Test Case 6	**The clerk request his officer for leave[1] but he ask him in anger to leave[2] the room.**	
Output Detail:	**Word: leave[1]** **Meaning:** to put (something) into the possession or safekeeping of another **Related words:** things, pen, book, books, chair, table, watch, possession, letter, coat, jacket, **Synonyms:** drop, hand in, give over.	Incorrect
	Word: leave[2] **Meaning:** to go away from **Related words:** room, office, place, ask, asked, tell, told, house, home, hours, minutes, alone, car. **Synonyms:** walk off, go away, elope, walk out, get away, move out.	Correct

We perform testing on different examples corresponding to our algorithm in which the words are to be disambiguated are written above. To measure the accuracy of our algorithm we perform testing on 6 lines. After testing we obtain 66% of accuracy as shown in Fig 4.
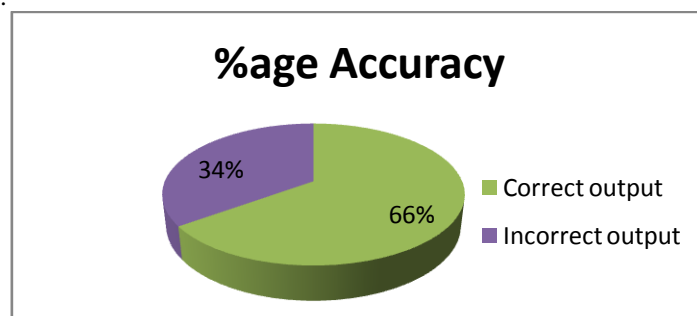

Fig 4: Histogram showing the accuracy of WSD Algorithm.

VI. Conclusion

Word sense disambiguation is a key problem to address in many applications in the areas of Natural Language Processing, Information Retrieval and others. The algorithm identifies the meaning of sentence like human brain. It disambiguates ambiguous words based on object on which sentence is written as in examples. In our proposed work we have developed an algorithm that can be used to disambiguate two polysemous words in a single line. These polysemous words can create hindrance for people to understand the accurate meaning of these words especially when they are occurred twice in a single line. In our work, the dictionary based approach is used. To measure the accuracy of our algorithm we perform testing on 100 lines. After testing we obtain 65% of accuracy. This can also be used to solve and test the problems related to English Grammar.

References

[1]   Rekha Jain and Neha Singh, "DISAMBIGUATION TECHNIQUE FOR POLYSEMOUS WORDS", Banasthali University, Jaipur, International Journal of Research in Engineering & Applied Sciences, Volume 2 Issue 2 ,February 2012.

[2]   Sabnam Kumari and Prof. (Dr.) Paramjit Singh, "Genetic Algorithm bases Hindi Word Sense Disambiguation", Department of Computer Science and Engineering, PDM College of Engineering, Bahadurgarh, Haryana, India, *IJCSMC, Vol. 2, Issue. 5, May 2013, pg.139 – 144.*

[3]   S. G. Kolte and S. G. Bhirud, "WordNet: A Knowledge Source for Word Sense Disambiguation", K. J. Somaiya Polytechnic, Mumbai-77, India, Veermata Jijabai Technological Institute, Mumbai – 19, India, International Journal of Recent Trends in Engineering, Vol 2, No. 4, November 2009.

[4]   Rada Mihalcea "Using Wikipedia for Automatic Word Sense Disambiguation" Department of Computer Science, University of North Texas, April 2007.

[5]   Jonas EKEDAHL, Koraljka GOLUB, "Word sense disambiguation using WordNet and the Lesk algorithm", Engineering Physics, Lund Univ. Tunav. 39 H537, 223 63 Lund, Sweden.

[6]   Satanjeev Banerjee, "Adapting the Lesk algorithm for Word Sense Disambiguation to WordNet", Master's thesis. Dept. of Computer Science, University of Minnesota, USA, 2002. http://www.d.umn.edu/~tpederse/Pubs/banerjee.pdf.

[7]   Word-sense disambiguation: http://en.wikipedia.org/wiki/Word-sense disambiguation.

[8]   Eneko Agirre, Word Sense Disambiguation: Algorithms and applications (springer site, book site) Series: Text, Speech and Language Technology, Vol. 33.

[9]   Ambiguity: http://plato.stanford.edu/entries/ambiguity/, *May 16, 2011.*

[10]   Rekha Jain and Neha Singh, "DISAMBIGUATION TECHNIQUE FOR POLYSEMOUS WORDS", Banasthali University, Jaipur, International Journal of Research in Engineering & Applied Sciences, Volume 2 Issue 2 ,February 2012.

[11]   Sabnam Kumari and Prof. (Dr.) Paramjit Singh, "Genetic Algorithm bases Hindi Word Sense Disambiguation", Department of Computer Science and Engineering, PDM College of Engineering, Bahadurgarh, Haryana, India, *IJCSMC, Vol. 2, Issue. 5, May 2013, pg.139 – 144.*

[12]   Xiaohua Zhou, Hyoil Han ,"Survey of Word Sense Disambiguation Approaches", College of Information Science & Technology, Drexel University 3401 Chestnut Street, Philadelphia, PA 19104, Appeared in The 18th FLAIRS Conference, Clearwater Beach, Florida, May 15-17, 2005.

[13]   Zhi Zhong and Hwee Tou Ng, "Word Sense Disambiguation Improves Information Retrieval", Department of Computer Science National University of Singapore 13 Computing Drive, Singapore 117417, *In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 273–282, Jeju, Republic of Korea ,July 2012.

[14]   Philip Resnik, David Yarowsky, "A Perspective on Word Sense Disambiguation Methods and their Evaluation", http://www.cs.jhu.edu/~yarowsky/pubs.html.

[15]   John, M., & Enss, R., "An Investigation of Word Sense Disambiguation for Improving Lexical Chaining", 2008.

[16]   English WordNet: http://WordNet.princeton.edu