Ambiguity Detection in Elementary Programming Problems

Himanshu Shukla, Kumar Gaurav
hshukla@iitk.ac.in, krgaurav@iitk.ac.in
IIT Kanpur

ABSTRACT

In this project we propose to develop a system for detecting ambiguities and anomalies in the elementary programming problems that are given in the introductory programming courses. Since ambiguity can occur because of both linguistic and contextual reasons, we will be using both the knowledge of linguistics and contexts of the problem to generate the results.

1 Motivation

In Programing, problem specification is a very important point especially if its an introductory programming course. For example , for the courses running in various universities and on various online learning mechanisms like MOOC, if the problem is ambiguous then the student ends up writing a wrong program code that takes a lot of time and energy as he/she is learning programming for the first time. Moreover it is also a headache for the instructor to keep on providing the clarifications over various issues in the problem raised by students. The problems are generally being tackled by sending clarification emails or posting in the forums whose result is not reflected immediately and also consumes a lot of time as it is not possible for the instructor to be on the portal everytime. Hence we want to design a model that takes the problem statement in english language from the instructor and shows any unambiguity which might be there. Such a system can be integrated within the online learning portals so that whenever the instructor posts a problem it will automatically tell him/her about the ambiguity for any necessary correction.

2 Data Set

Treebanks and corpus are available for general english language but not for the programming problems. The types of problems is available at SigPact IITK (ESC-101) programing platform and also at SPOJ (beginners level). We will build the training data set by asking various people(students, TAs etc.) of IIT Kanpur. The data obtained would be manually parsed into suitable format for training.

3 Implementation

The problems statements received from data collection would be classified into ambiguous and unambiguous categories. Stanford Lexicalized parser [1–3] and Coreference resolution [4–7] tool would be used to get a dependency parse tree and get mathematical objects that are mentioned in the statement. A model will be built that would create a relationship graphs between the attributes of the mentioned objects and if they are enough to specify the object correctly. The relationship can be found using Kernel Methods [8] from sentences that are already resolved from coreferences and anaphora in the previous steps. Cases where attributes are not enough or may lead to two or more different objects would constitute an unsolvable or ambiguous statement respectively. The model would be trained using the manually obtained dataset.

4 Challenges

- One of the biggest challenges will be the collection of data which is dependent on the fact how well the people coordinate.
- Second big challenge would be the model creation for various domains.
- Third challenge will be of defining a kernel [8] so that we can project the data in much larger dimension.

References

[1] Fundel, K., Küffner, R., and Zimmer, R., 2007. "Relexrelation extraction using dependency parse trees". *Bioinformatics,* **23**(3), pp. 365–371.

[2] De Marneffe, M.-C., MacCartney, B., Manning, C. D., et al., 2006. "Generating typed dependency parses from phrase structure parses". In Proceedings of LREC, Vol. 6, pp. 449–454.

[3] Socher, R., Bauer, J., Manning, C. D., and Ng, A. Y., 2013. "Parsing With Compositional Vector Grammars". In *ACL*.

[4] Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., and Jurafsky, D., 2013. "Deterministic coreference resolution based on entity-centric, precision-ranked rules". *Computational Linguistics,* **39**(4), pp. 885–916.

[5] Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., and Jurafsky, D., 2011. "Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task". In Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, pp. 28–34.

[6] Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., and Manning, C., 2010. "A multi-pass sieve for coreference resolution". In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 492–501.

[7] Recasens, M., de Marneffe, M.-C., and Potts, C., 2013. "The life and death of discourse entities: Identifying singleton mentions.". In HLT-NAACL, pp. 627–633.

[8] Zelenko, D., Aone, C., and Richardella, A., 2003. "Kernel methods for relation extraction". *The Journal of Machine Learning Research,* **3**, pp. 1083–1106.