Linguistic Pragmatics Theory in Natural Language Processing for Games

Zooraze Tariq — Daniel Chunn — Karissa Tuason — Andrew Cai

Abstract

The gameplay will consist of various semi-autonomous agents that the player must convince, command, and cajole into performing the assigned tasks. Tasks will start simply and grow in difficulty. The player will be able to communicate clearly and efficiently in order to win the game faster than with conventional controls. These tasks can be small puzzles or other increasingly complex objectives.

We hope to study Linguistic Pragmatics Theory and Natural Language Processing through crowdsourcing, creating a more immersive experience through less repitition and forcing and more elegant text processing for understanding.

1 Introduction

Problem Statement

Linguistic pragmatics in natural language for game dialogue systems for better human-computer interaction.

Motivation

Recurrent neural networks and linguistic pragmatics theory has yet to make its way into dialogue in games or even very much in artificial intelligence on a whole. We hope to allow players to communicate with computers to efficiently and effectively solve puzzles using their native language prowess.

Application

There are various applications for such a system, including but not limited to training players to communicate efficiently and improve grammar and speech skills, navigate an operating system, browse the web, etc. We will focus on applying it to making gameplay more natural and fluid.

Challenges

1. Recurrent Neural Network
2. Linguistic Pragmatics with RNN
3. Parsing for understanding
4. Models based on discourse representation theory
5. Creating human readable discourse structures
6. Recognizing and actioning command patterns

Prior Work

Previously the problem of natural language processing for understanding tends to be tackled with brute force, such as in the case of IBM's Watson. Other work tends to focus on applications in search heuristics and planning problems.

Since there is currently no work being done on the problem for the same application, we are leveraging work of a similar nature.

Proposed Solution

We have split the problem into two parts: text processing and generation.

For the generation we will use an RNN backed by a teaching engine based in linguistic pragmatics theory. The parser will implement algorithms inspired by discourse representation structure and optimized using scope underspecification.

The benefit of this approach is more efficient processing over time and, post processing, more organized, human-readable data which can be understood by non-experts and further utilized for crowdsourcing, study and practical applications.

Contributions

1. RNN using Linguistic Pragmatic Models
2. Discourse Representation Structure for NLP parser
3. Scope Underspecification for optimization of real-time parsing
4. Graph-based macro system
5. Command pattern recognition and visualization system

Benefits.

1. More efficient processing of commands over time
2. Fluid human-computer interaction
3. Organized, human-readable data which can be understood by non-experts

Main Result / Deliverable

A game which will consist of various semi-autonomous agents that the player must convince, command, and cajole, through natural language alone, into solving increasingly challenging puzzles.

Demonstration

Metrics (what performance measures will be recorded): Speed of completion, accuracy in sentence construction, vocabulary size, and reported confidence in survey.

We will measure performance in the tasks over time and response to simple skill tests. We will do the analysis partially with the algorithm in this paper, "The Moving-Average Type–Token Ratio (MATTR)" for vocabulary strength. We will analyze accuracy of grammar by simple syntax matching.

Evaluation.

1. Quantitative Evaluation: Speed of completion, accuracy in sentence construction, vocabulary size, and reported confidence in survey. MATTR, Syntax Match Percentage
2. User Study: Players will complete puzzles, take and give feedback to the AI, then take a survey reporting their confidence.
3. Qualitative Evaluation: Video demos, Powerpoint, Graphs

2 Related Work

Linguistics

1. The work in [Schlenker 2008] covers pragmatics theory of presupposition projection.
2. File change semantics and the familiarity theory of definiteness in [Heim 1983].

Recurrent Neural Networks and Parsing

1. This should be a quality improvement to generated output (at the cost of adding a constant multiplier to the running time) and seems easy enough to implement on top of an RNN [Graves 2012].
2. RNN optimization, application,and difficulties, including Hessian-free optimizer and optimal control, and Restricted Boltzmann Machines [Sutskever 2013].

Smart Objects

1. Applies a BT to create a Starcraft bot with multiple goals. This system might be too complex to implement [Weber et al. 2010].

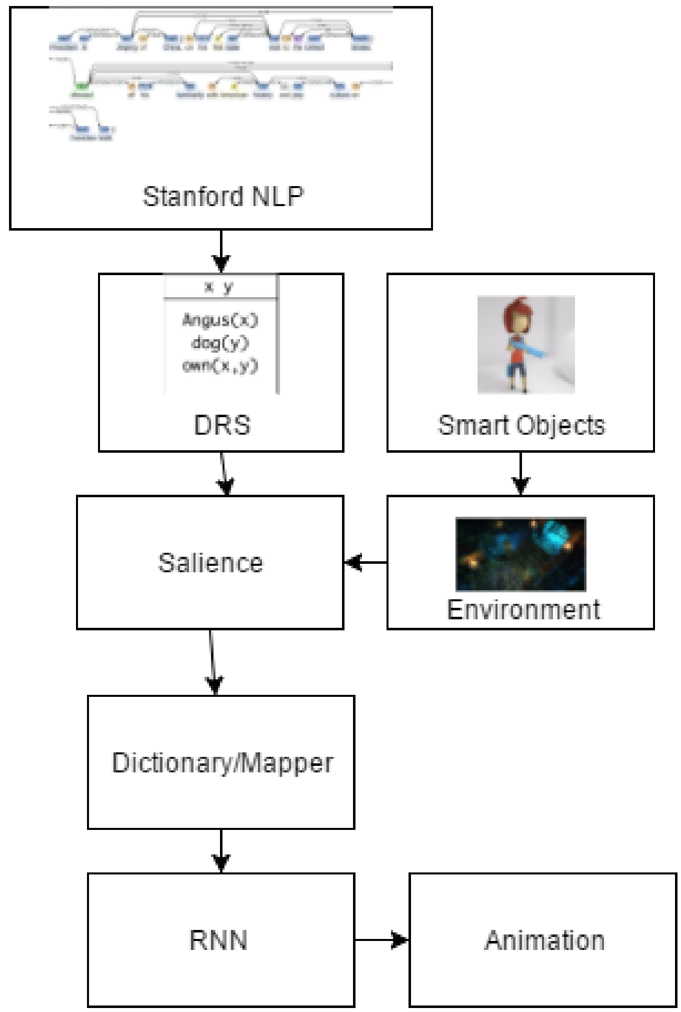
Miscellaneous

1. Optimization technique for machine learning [Botea et al. 2005].
2. Overview of deep learning for web search and natural language parsing [Gao et al. 2015].
3. Two deep parsing components, an English Slot Grammar (ESG) parser and a predicate-argument structure (PAS) builder, provide core linguistic analyses of both the questions and the text content used by IBM's Watson to find and hypothesize answers. [McCord et al. 2012].
4. The aim of this work is to exploit the conventional NER methods for analyzing a large set of microtexts of which lengths are short. This is essentially a case study of Twitter [Jung 2012].
5. This study designed a chatbot system, Confucius, as a MSN virtual learning companion to examine how specific application design variables within educational software affect the learning process of subjects as defined by the cognitive continuum of field-dependent and field-independent learners [Hsieh 2011].
6. This studies objective is to provide robust understanding of complex requests while giving the user flexibility in their language. [Booth et al. 2015].
7. An overview of artificial intelligence [Russell et al. 1995].
8. Sentiment strength detection from informal text [Thelwall et al. 2010].
9. Automatic learning and generation of social behavior from collective human gameplay [Orkin and Roy 2009].
10. A real-time learning control approach for nonlinear continuous-time system using recurrent neural networks [Chow et al. 2000].

Comparison to Prior Work.

The problem prior work has brought to light is to facilitate real-time dialogue between the player and the NPCs, to allow the NPCs to react, and to create appropriate sentences to relay feedback in a natural manner. To this end we plan to employ statistical natural language processing by implementing parts of the OpenNLP and StanfordNLP library in Unity and implementing sentiment recognition using linguistics pragmatic theory, and identifying situational domains (persons, objects, and events of interest). To have our computer recognize and understand language, we will implement algorithms based in pragmatics theory, such as discourse representation structures, file change semantics, and graded salience.

3 Framework



4 Implementation Details

4.1 Discourse Representation Structure

Discourse Representation Structure (DRS) provides an efficient representation of the input sentence to allow for well organized and easy to use data. Our current implementation uses StanfordNLP as a basis.

4.2 Salience

We plan on implementing Salience for resolution of amiguous references in players' input. Salience is a measure of how important a person would consider something to be. For each smart object, NPCs will keep track of various measures of relevance, such as how frequently or when it was last mentioned. The NPC's field of view, spatial distance and inherent salience for that object will also be calculated and used. The relative importance of these measures can vary between NPCs, making each more unique while also removing the need to adjust this precisely.

4.3 RNN

The RNN uses the SharpML.Recurrent library which is a C Sharp implemetation of Andrej Karpathy's RecurrentJs and Thomas Lahore's RecurrentJava.

Currently the RNN model training is too computationally intensive to run in Unity, so we are doing training on a console application with suboptimal parameters. In Unity we've implemented a simple interface to test the pre-trained model and voabulary list, and output some predicted words. Even though we'll be able to swap our test model and vocabulary list with better ones, we plan to modify the RNN in order for future inputs to be stored and trained for the RNN.

The RNN is trained using a word-level model rather than the character-level model found in famous RNN examples like RNN-Bible. This makes each word an input/output and each sentence a sequence and dependency to learn.

5 Results and Evaluation

5.1 Results

We are experiencing some difficulties with certain words being incorrectly identitfied by StanfordNLP, for instance the word "attack" may be labeled as an noun dispite it being used as a verb.

5.2 Computational Performance

While running the models we are using is sufficiently efficient for real time use, the loading of large model files can take over a minute on slower machines. This does not impact the game experience itself, but players may dislike the long load times.

5.3 Quantitative and Qualitative Evaluation

Based on qualitative data, we have made some progress in providing more natural communication with computers. We do not yet have quantitative data but are working to obtain some in our second user study.

6 User Study

6.1 Method

Our hypothesis is that players will be able communicate more efficiently with the computer using natural language and solve puzzles faster than when using traditional input systems.

As our dependant variable, we have the time it takes for a player to solve a puzzle. Our independant variables are two different groups playing through the same puzzle using different methods of input. One group uses language to communicate with the computer while the other group uses the traditional method of keyboard and mouse. We hope to see that using language is a more effecient way to solve puzzles.

6.2 Questionnaire

Before we have players play the game, we ask a few questions before hand such as their previous experiences with computer interaction and what kind of language format they expect to use to communicate with computers.

After the players play the game, we have another series of questions such as what type of language they used to communicate with the computer and what improvements can be made to make interacting with the computer easier.

6.3 User Study

In our pre-game study, we found that players use and expect very syntactically strict language when communicating with computers. This includes command line arguments and simple text-based games that utilize simple language. Players consider interaction with computers to be very different from human interaction. On a scale of 1-5, with 1 being very similar and 5 being very different from human interaction, the average rating the players gave was a 4.66.

In our post-game study, some players found that there was some freedom to their input, allowing for some more natual language to be used. However, some players did find that their inputs had no effect at all and had hoped that the computer would be able to accept mistakes or ambiguities. Overall, the average rating in the difference between computer interaction and human interaction was a 3, meaning that computer interaction was a little more natural than what the players had expected.

7 Conclusion

7.1 Limitations

Currently our implementation does not allow sufficient room for ambiguities or mistakes, making the language required still somewhat precise. Players had also found that the computer just does not support enough actions, which made the game they've played to be very minimalistic and barebones.

7.2 Future Work

We hope to add more content to our game in the future. This includes accepting and understanding more input commands, such as questions, ambiguous commands, and mistakes. Other features we wish to implement include more supported actions and puzzles. Actions such as moving freely as opposed to moving to a specific destination, unlocking doors and chests, and combat. Implementing more actions allows us to create more fun and diverse puzzles that will help grow our game.

References

BOOTH, J., DI EUGENIO, B., CRUZ, I. F., AND WOLFSON, O. 2015. Robust natural language processing for urban trip planning. *Applied Artificial Intelligence 29*, 9, 859–903.

BOTEA, A., ENZENBERGER, M., MÜLLER, M., AND SCHAEFFER, J. 2005. Macro-ff: Improving ai planning with automatically learned macro-operators. *Journal of Artificial Intelligence Research*, 581–621.

CHOW, T. W., LI, X.-D., AND FANG, Y. 2000. A real-time learning control approach for nonlinear continuous-time system using recurrent neural networks. *Industrial Electronics, IEEE Transactions on 47*, 2, 478–486.

GAO, J., HE, X., AND DENG, L. 2015. Deep learning for web search and natural language processing. *Microsoft Research Technical Report*.

GRAVES, A. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

HEIM, I. 1983. File change semantics and the familiarity theory of definiteness.

HSIEH, S.-W. 2011. Effects of cognitive styles on an msn virtual learning companion system as an adjunct to classroom instructions. *Educational Technology & Society 14*, 2, 161–174.

JUNG, J. J. 2012. Online named entity recognition method for microtexts in social networking services: A case study of twitter. *Expert Systems with Applications 39*, 9, 8066–8070.

MCCORD, M. C., MURDOCK, J. W., AND BOGURAEV, B. K. 2012. Deep parsing in watson. *IBM Journal of Research and Development 56*, 3.4, 3–1.

ORKIN, J., AND ROY, D. 2009. Automatic learning and generation of social behavior from collective human gameplay. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, International Foundation for Autonomous Agents and Multiagent Systems, 385–392.

RUSSELL, S., NORVIG, P., AND INTELLIGENCE, A. 1995. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs 25*, 27.

SCHLENKER, P. 2008. Be articulate: A pragmatic theory of presupposition projection. *Theoretical Linguistics 34*, 3, 157–212.

SUTSKEVER, I. 2013. *Training recurrent neural networks*. PhD thesis, University of Toronto.

THELWALL, M., BUCKLEY, K., PALTOGLOU, G., CAI, D., AND KAPPAS, A. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology 61*, 12, 2544–2558.

WEBER, B. G., MAWHORTER, P., MATEAS, M., AND JHALA, A. 2010. Reactive planning idioms for multi-scale game ai. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, IEEE, 115–122.