**AIA: Assassin Handler, Evaluation of Natural Language Processing**

Natalia Porras

*School of Computing Teesside University N3269117@live.tees.ac.uk*

Stephen Sawyer

*School of Computing Teesside University N3033749@live.tees.ac.uk*

Oliver Hickey

*School of Computing Teesside University N3263953@live.tees.ac.uk*

The Abstract

State the problem, your approach and solution, and the main contributions of the paper. Include little if any background and motivation. Be factual but comprehensive. The material in the abstract should not be repeated later word for word in the paper.

(**Exercise:** Write an abstract for the multiway sort example.)

This paper will discuss Language Processing, on how some aspect of semantically-based language processing can be achieved in the context of a system which aims to infer appropriate meanings for non-trivial sentences, this is an extension of ‘Methodologies of an Assassin, Evaluation Different Inference Engines’. Clojure and Netlogo will be primarily used for this Experiment, the goal is to be able to process language into the Assassin Scenario and then see how the system reacts to it.

The Introduction

The Introduction is crucially important. By the time a referee has finished the Introduction, he's probably made an initial decision about whether to accept or reject the paper -- he'll read the rest of the paper looking for evidence to support his decision. A casual reader will continue on if the Introduction captivated him, and will set the paper aside otherwise. Again, *the Introduction is crucially important.*

Here is the [Stanford InfoLab](http://infolab.stanford.edu/)'s patented five-point structure for Introductions. Unless there's a good argument against it, the Introduction should consist of five paragraphs answering the following five questions:

1. *What is the problem?*
2. *Why is it interesting and important?*
3. *Why is it hard?* (E.g., why do naive approaches fail?)
4. *Why hasn't it been solved before?* (Or, what's wrong with previous proposed solutions? How does mine differ?)
5. *What are the key components of my approach and results?* Also include any specific limitations.

The purpose of this Paper and model is to display and experiment how Language Processing reacts with the system that was built for the last paper. This system contains the Assassin Model that was designed and developed for the last paper, from this the language processing method will be built on top of that, so the system that will react to the language processing will already be in effect, therefore the language processing should be able to yield the results necessary.

This Language processing will be designed and developed using Netlogo and Clojure.

The Natural Language Processing is an AI method so you can communicate with intelligent programs and system using natural languages like English. The Processing of Natural Language is required to be set up when you want the intelligent system to perform or react to your instructions. Being able to communicate to your system using language, is a good way of seeing how well your AI system performs and functions, because if you are giving exact instructions in English and it replicates your instructions, then it shows how far you have come with it.

Although there is difficulties with Lexicon system that is being purposed for the system that has been created. One of the biggest problem that is faced with the system, is within the English language, is that different words have different meanings and have different connotations. For example; you can the word Park, park can be a noun or a verb at the same time, so if you try communicate with a AI system, it may have difficulties understanding which version of the word you mean, therefore causing it to completely disregard what you said or try and attempt what it assumed you meat.

Related Work

The Body

**Guideline #1:** A clear new important technical contribution should have been articulated by the time the reader finishes page 3 (i.e., a quarter of the way through the paper).

**Guideline #2:** Every section of the paper should tell a story. (Don't, however, fall into the common trap of telling the entire story of how you arrived at your results. Just tell the story of the results themselves.) The story should be linear, keeping the reader engaged at every step and looking forward to the next step. There should be no significant interruptions -- those can go in the Appendix; see below.

Aside from these guidelines, which apply to every paper, the structure of the body varies a lot depending on content. Important components are:

* **Running Example:** When possible, use a running example throughout the paper. It can be introduced either as a subsection at the end of the Introduction, or its own Section 2 or 3 (depending on Related Work).
* **Preliminaries:** This section, which follows the Introduction and possibly Related Work and/or Running Example, sets up notation and terminology that is not part of the technical contribution. One important function of this section is to delineate material that's not original but is needed for the paper. Be concise -- remember Guideline #1.
* **Content:** The meat of the paper includes algorithms, system descriptions, new language constructs, analyses, etc. Whenever possible use a "top-down" description: readers should be able to see where the material is going, and they should be able to skip ahead and still get the idea.

Performance Experiments

We could have an entire treatise on this topic alone and I am surely not the expert. Here are some random thoughts:

* Many conferences expect experiments.
* It's easy to do "hokey" or meaningless experiments, and many papers do.
* It's easy to craft experiments to show your work in its best light, and most papers do.
* What should performance experiments measure? Possiblities:
  + Pure running time
  + Sensitivity to important parameters
  + Scalability in various aspects: data size, problem complexity, ...
  + Others?
* What should performance experiments show? Possibilities:
  + Absolute performance (i.e., it's acceptable/usable)
  + Relative performance to naive approaches
  + Relative performance to previous approaches
  + Relative performance among different proposed approaches
  + Others?

The Conclusions

In general a short summarizing paragraph will do, and under no circumstances should the paragraph simply repeat material from the Abstract or Introduction. In some cases it's possible to now make the original claims more concrete, e.g., by referring to quantitative performance results.

Future Work

This material is important -- part of the value of a paper is showing how the work sets new research directions. I like bullet lists here. (Actually I like them in general.) A couple of things to keep in mind:

* If you're actively engaged in follow-up work, say so. E.g.: "We are currently extending the algorithm to... blah blah, and preliminary results are encouraging." This statement serves to mark your territory.
* Conversely, be aware that some researchers look to Future Work sections for research topics. My opinion is that there's nothing wrong with that -- consider it a compliment.

The Acknowledgements

Don't forget them or you'll have people with hurt feelings. Acknowledge anyone who contributed in any way: through discussions, feedback on drafts, implementation, etc. If in doubt about whether to include someone, include them.

Citations

Spend the effort to make all citations complete and consistent. Do *not* just copy random inconsistent BibTex (or other) entries from the web and call it a day. Check over your final bibliography carefully and make sure every entry looks right.

Appendices

Appendices should contain detailed proofs and algorithms only. Appendices can be crucial for overlength papers, but are still useful otherwise. Think of appendices as random-access substantiation of underlying gory details. As a rule of thumb:

* Appendices should not contain any material necessary for understanding the contributions of the paper.
* Appendices should contain all material that most readers would not be interested in.