- URL: http://www.ijiee.org/papers/280-N011.pdf

- Title: An Efficient Network Monitoring and Management System

- Authors: Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Muhammad Inayatullah Babar

- Notes: This describes using Nagios in detail, which could be INCREDIBLY useful later because it turns out Nagios has a JSON API. My chat bot + Nagios API could be an awesome way of getting incredibly detailed information to the admin. Worth exploring.

##Review

According to Khan et al. 2013[1], large organizations require fast and efficient network monitoring systems that reports to a network administrator via email or SMS as soon as a problem arises, with details of the problem and locations affected. They go on to explain the merits of Nagios, a network monitoring tool, and its role in their system. It is extremely important, they mention, that the system be essentially autonomous in operation, as in a large company manual monitoring is very difficult.

The paper contains some basic instruction for configuring a Nagios setup and defines several ways for the software to check the status of various servers and services, and could act as a good guide for somebody new to Nagios configuration. However, the paper does not explore additional ways of informing administrators of issues, nor does it compare other software that may have similar features.

The method used to set up Nagios means that their system interacts with a Request Tracker, and as Nagios detects faults in the network it will send affected nodes plus other information via the Request Tracker as a ticket to the network admin. If the ticket isn’t resolved in an hour, the ticket is send to the second responsible network person. This method means that all persons are informed one by one until the ticket is marked as resolved. This could mean some dissonance may exist at times between different responsible persons, but this is not addressed in the paper.

###Citation

Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and M. I. B. (2013). An Efficient Network Monitoring and Management System. *International Journal of Information and Electronics Engineering*, *3*(1), 122. https://doi.org/10.7763/IJIEE.2013.V3.280

- Url: http://www.jmlr.org/papers/volume12/collobert11a/collobert11a.pdf

- Title: Natural Language Processing (Almost) from Scratch

- Authors: Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukeuoglu, Pavel Kuksa

- Notes: Jason Weston was a research scientist at NEC Labs, Google and Facebook. That's some freaking pedigree right there.

##Review

No consensus has emerged whether a piece of software will ever be able to convert English text into a programmer friendly data structure that describes the meaning of the text, according to a paper by Collobert et al. 2011[2]. The paper was written to accompany an attempt to build a natural language parser using a huge database of training data, and documenting the process of machine learning. Their approach was benchmarked using four standard NLP tests:

* Part of Speech tagging – This aims at labelling each word with a tag that indicates its *syntactic role*, such as plural, noun, adverb etc.
* Chunking (or shallow parsing) – This aims at labelling segments of a sentence with syntactic constituents such as noun or verb phrases, where each word is assigned a tag and encoded as a ‘begin-chunk’ or ‘inside-chunk’ tag.
* Named Entity Recognition – This aims to label elements into categories such as “PERSON” or “LOCATION”.
* Semantic Role Labelling – This aims at giving a semantic role to a syntactic constituent of a sentence.

Their experiments followed the standard evaluation procedure of the CoNLL challenges, a set of tasks with the goal of challenging the computer science community to create machine learning strategies which address proposed natural language processing problems.

The paper criticizes itself, noting that they used multilayer neutral networks, a 20-year-old technology, rather than something more modern, though they also note that the training algorithm used was only possible due to the tremendous progress in computer hardware. Due to their unique approach of trying to build from scratch rather than using work already established, much potentially relevant information from other papers and previous experiments could be construed as missing.

###Citation

Collobert, R., Weston, J., & Bottou, L. (2011). Natural language processing (almost) from scratch. *The Journal of Machine …*, *12*, 2493–2537. <https://doi.org/10.1145/2347736.2347755>

- URL: <http://researchbriefings.parliament.uk/ResearchBriefing/Summary/POST-PN-389>   
- Title: Cyber Security in the UK  
- Authors: Chandrika Nath  
- Notes: Potentially not very relevant, does it even count as a paper?

##Review

The Cyber Security in the UK paper[3] explains, in detail, the British governments approaches to cyber security. It also describes various different types of attacks, such as data theft, attacks on critical information infrastructure, and attacks on physical infrastructure. The paper does well to inform the reader of terminology related to the field, such as the concept of air-gapping – network isolation, and zero-day attacks – previously unknown and unprotected attacks.

The paper suggests that common cyber security measures include methods such as the deployment of firewalls, using up to date anti-virus software, regular software patching, access management, encryption, and use of intrusion detection software. It also stresses the importance of security in industrial control systems, such as smart metering of gas and electricity consumption in homes, to avoid data falsification or damage to systems.

While there are some examples of major high profile attacks such as the Stuxnet virus and the data thefts at Lockheed Martin, the details of these attacks are very lacking and there are few sources to follow for more information. The paper offers no real scrutiny or analysis, and merely informs the reader, where it would have been nice to see a comparison between other countries cyber security plans.

##Citation

Chandrika, N. (2011). Cyber Security in the UK. *POSTnote*, (389), 1–4. Retrieved from http://www.parliament.uk/business/publications/research/briefing-papers/POST-PN-389

- URL: <http://ieeexplore.ieee.org/document/5375542/>   
- Title: Visualising Cyber Security: Usable Workspaces  
- Authors: Glenn A. Fink ; Christopher L. North ; Alex Endert ; Stuart Rose  
- Notes: Apparently command line tools are primitive, so we should all use GUI’s instead.

##Review

‘Visualizing Cyber Security: Usable Workspaces’[4] provides an insight into an experiment involving adding more visualisations into cyber security, and phasing out ‘primitive’ command line tools. They replaced eight cyber analysts screen setups with a 4x2 configuration and recorded results when faced with generic visualisations of Net-flow and Snort alert data, which were met with mixed opinions.

The paper quotes the opinions of many analysts, even stating broadly that cyber analysts in general dislike visualisations, and prefer command line due to its flexibility and expressive power. While they did find several situations where visualisations helped in finding complex patterns in real world data, they found it very hard to sell the idea of visualisation to seasoned analysts. This distrust may stem from poor performance of instruction detection systems, which attempt to automate and “simplify” the process, as the number of false positives emitted is supposedly very high.

Much of the experimentation done in the paper was met with harsh response by the analysts, with many comments such as their original approaches, using grep or SQL queries, being considerably faster than the visualised equivalent. These comments are met with unconvincing defences by the paper, stating that is was not a fault of the visualisation tool in use, but rather bad database management. The paper could have spent much more time discussing possible alternatives rather than trying to defend its choices when met with criticism.

##Citation

Fink, G. A., North, C. L., Endert, A., & Rose, S. (2009). Visualizing cyber security: Usable workspaces. In *6th International Workshop on Visualization for Cyber Security 2009, VizSec 2009 - Proceedings* (pp. 45–56). https://doi.org/10.1109/VIZSEC.2009.5375542

- URL: <http://www.ijser.org/researchpaper/Study-of-Latest-Emerging-Trends-on-Cyber-Security-and-its-challenges-to-Society.pdf>   
- Title: Study of Latest Emerging Trends on Cyber Security and its challenges to Society  
- Authors: Ravi Sharma  
- Notes: Focuses on security emerging trends while adopting new technologies. Bachelor of engineering with a Hotmail email address, possibly an idiot.

##Review

A study in the *International Journey of Scientific & Engineering Research*[5] shows new trends in cyber security based on the adoption of new technologies, claiming that due to the Windows 8 and onwards unified architecture between devices, attacks will be easier than ever between a range of systems. It also claims that due to this, somehow it would be possible to develop malicious applications like those for Android.

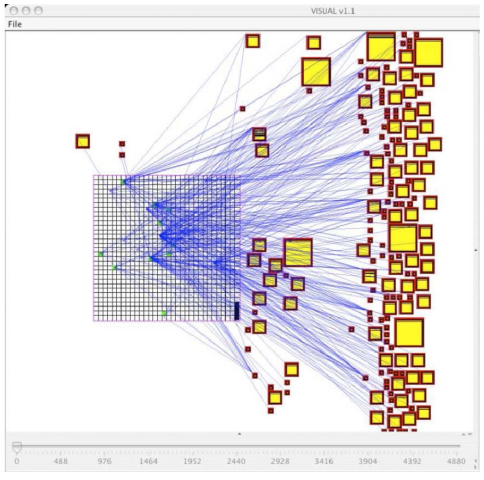
The study makes numerous claims but a good number seem to be not fully backed up by their referenced media, or lacks any sort of study that can be referenced at all. The study also claims via its abstract to discuss lack of coordination between security agencies and critical IT infrastructure, though this was not covered in detail.

##Citation

Sharma, R. (2012). Study of Latest Emerging Trends on Cyber Security and its challenges to Society. *International Journal of Scientific & Engineering Research*, *3*(6). Retrieved from http://www.ijser.org

- URL: <http://ieeexplore.ieee.org/document/6007132/>   
- Title: A Survey of Visualisation Systems for Network Security  
- Authors: H. Shiravi, A. Shiravi, A.Ghorbani  
- Notes: Also includes me fiddling with the new in-line citation style. Has a super-useful table of where information can come from, could be good to link to later when looking at security events from IDS and how to format data.

##Review

Security Visualization is a very young term, and many common visualisation techniques are not designed for security related data [6]. Manually traversing textual logs is not only frustrating and time consuming, but may result in important details being overlooked. This paper explores methods of showing administrators quantitative data in meaningful ways so as to better look for anomalies or patterns from sources such as intrusion detection systems, port scanning tools and firewalls.

The paper shows interesting ways of converting vast quantities of event types into graphs and other visual representations. This included splitting the events into different types: Network traces, security events, network activity context, user/asset context, network events and application logs. The image to the right shows 80 hours of network data on a network of 1020 hosts. The internal network is represented by the grid on the left, and external servers by squares on the right, wit square size denoting the level of activity. This is not extremely obvious upon first look.

One thing the paper does not do is really discuss the requirements of visualisation compared to merely parsing and reformatting text data. While the image above does a good job of showing that large amounts of data may be transferred to external servers from internal sources, we have no easy way of knowing if any of this data is malicious or coming from potentially unwelcome sources. If this data were formatted as an easily formatted excel table for example, we could apply filters very easily to look for patterns in data. This is a common theme in the paper, where visualisations show what at first seems to be useful information, but allows little to no exploration of potential anomalies once identified short of digging through raw data.

##Citation

Shiravi, H., Shiravi, A., & Ghorbani, A. a. (2012). 006 A survey of visualization systems for network security. *IEEE Transactions on Visualization and Computer Graphics*, *18*(8), 1313–1329. https://doi.org/10.1109/TVCG.2011.144

- URL: <http://jamia.oxfordjournals.org/content/jaminfo/18/5/544.full.pdf>   
- Title: Natural Language Processing: An Introduction  
- Authors: Prakash M Nadkarni, Lucila Ohno-Machado, Wendy W Chapman  
- Notes: Aimed at a medical audience but could still be relevant as an NLP intro

##Review

*Natural Language Processing: An Introduction*[7] acts as an overview of common machine-learning approaches currently being used and possible future directions of NLP, as well as some of the associations with IR – Information Retrieval. One of the first things defined is that of statistical NLP –NLP based on machine learning methods, learning via large annotated bodies of text which provided the standard they were looking to achieve. The paper quickly becomes complex, looking into data driven approaches to NLP and their drawbacks such as Hidden Markov Models (HMMs), which is a system where variables can switch between several states and generate possible outputs. The issue with HMM’s is that we can only see the output, not the process at it takes to comes to that output.

While this paper is primarily looking into NLP, it more looks into the methods of machine learning that could be applied to NLP. However, it still lists and explains a good number of NLP sub-problems which, while primarily aimed toward the medical field, are still relevant in computer science, such as sentence boundary detection and morphological decomposition – The act of separating words into smaller words.

The paper also has an interesting section focussing on the future of artificial intelligence and NLP, quoting heavily IBM’s Watson supercomputer and its attempt at beating humans in the game Jeopardy. With 16TB ram, Watson is designed to hold all of reference content in memory, as opposed to being disk-I/O-bound, which makes its seek time exceptionally slow. However, Watson can be easily misled with certain questions – Asking it “Which US city has two airports, one named after a World War II Battle, the other after a World War II Hero?” would be a multi-step process which Watson could not answer, as the reference content used for machine learning was structured as one sentence question and answers (“What/who is/are X?”).

##Citation

Nadkarni, P. M., Ohno-Machado, L., Chapman, W. W., Manning, C., Raghavan, P., Schuetze, H., … Miller, R. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association : JAMIA*, *18*(5), 544–51. https://doi.org/10.1136/amiajnl-2011-000464

- URL: http://www.alicebot.org/anatomy.html  
- Title: The Anatomy of A.L.I.C.E  
- Authors: Wallace, Richard S  
- Notes:

##Review

The Anatomy of A.L.I.C.E[8] is a paper that explains and represents the technical side of the Artificial Linguistic Internet Computer Entity, the winner of the Loebner Prize as “the most human computer”. The paper goes into detail on Turing’s 1950’s paper on the Original Imitation Game (OIG) where a man and woman are asked questions remotely, via text, by an interrogator. The man is instructed to lie and ensure the interrogator is not able to find out that the man is a woman. Turing proposes replacing the man with a robot, and so came the description of the Turing Test.

The paper goes on to criticise ALICE’s winning of the Loebner Prize, as it is purely designed to beat the Turing Test and is not designed for use in real world applications. ALICE consists of a huge database of AIML (XML style) elements, each combining questions and answers or stimulus and responses in an attempt to match any potential question with a reasonable response.

In this way, ALICE is based on the original ELIZA program, an earlier natural language processor in development from 1964-1966. The original ELIZA program was released under the guise of ‘Doctor’ for use by nontechnical staff in MIT. The issue with this was that nontechnical staff thought that this ‘Doctor’ was a real therapist and spent hours revealing personal problems to a script that merely either rewrote what they had said with the pronouns reversed, or answered with pre-prepared statements based on simple pattern matching. This, at the time, was thought to be rather dangerous.

The AIML elements are designed to symbolically reduce sentences into simpler formats to understand, or match elements and patterns. For example, ALICE programmers preferred to reduce sentences to the simplest possible form, which would be referred to as the template.

<category>  
<pattern>DO YOU KNOW WHO \* IS</pattern>  
<template><srai>WHO IS <star/></srai></template  
</category>

‘SRAI’ allows AIML to commit recursion to further simplify patterns.

It then attempts to “divide and conquer” sentences by reducing it to subsentences. For example, if a sentence begins with “Yes” (to answer a question), and then has more words, then both sections would be treated as subsentences as go down separate paths in the AIML tree.

The paper talks at length at how the ALICE bot continues to try and build its responses from the context of conversation, including previous responses, and how it is capable of remembering pronoun bindings using predicates. However, the paper does not conjecture on possible improvements to the ALICE bot nor what other works may be in process. It also does not describe the computational requirements or any examples of how the code that runs the AIML works, which is disappointing.

##Citation

Wallace, R. S. (n.d.). The Anatomy of A.L.I.C.E. Retrieved from http://www.alicebot.org/anatomy.html

- Title: Chatbots: Are they really useful?  
- Authors: Atwell, Eric; Shawar, Bayan Abu  
- Notes:

##Review

*Chatbots: Are they really useful?*[9]is essentially a review of many different chat bots and their uses to date, many of which use AIML elements and are based on ALICE. It has basic explanation of ALICE’s pattern matching algorithm and how it deals with incoming input. For example, all input has punctuation removed, is split into multiple sentences as needed, and then will try and match word by word to obtain the longest pattern match, which would be expected to be the best one. It will then use those matched patterns to determine which templated response to use.

The current public-domain ALICE “brain” has been built up by Dr Richard Wallace, and contains more than 50,000 categories of possible patterns, but all these categories were hand coded, which is extremely time consuming if one were interested in building a chat bot for themselves. To combat this idea, the paper also includes information about a Java program that attempts to automatically turn text into AIML style scripting. However, this was found to not be satisfactory to trial users.

The paper also discusses Pandorabot, a website used to build and deploy chat bots using the AIML XML system, which could be useful.

While the paper does include some original work, much of the reviews and comparisons of the chatbot systems are references to the same author, and there is little evidence from other authors. The paper also rarely reflects upon bots that do not use the AIML system.

##Citation

Shawar, B. A., & Atwell, E. (2007). Chatbots: are they really useful? *LDV-Forum*.

- Title: Using dialogue corpora to train a chatbot  
- Authors: Shawar, Bayan Abu; Atwell, Eric  
- Notes:

##Review

*Using Dialogue Corpora to Train a Chatbot* presents two chatbot systems, ALICE and Elizabeth and attempts to compare and pattern matching techniques of each system. Using the Dialog Diversity Corpus, a database of formatted human texts that can be used for human interaction research, the researchers attempt to convert natural language into AIML format.

After a section explaining symbolic reduction and how the AIML format handles recursion, which interestingly seems to be taken straight from the original paper by Dr Richard Wallace, with no direct reference, the paper moves on to the Elizabeth chat bot.

Elizabeth is a much-improved adaptation of the Eliza program, but much simpler in terms of database management than ALICE. Elizabeth uses a single script to run the bot, and may contain at most four parts. The first part deals with welcoming the user, and what to do in the case of an empty message, or a message which it does not understand.

The second part of the script deals with transformations, such as “MUM -> mother”. This can be seen as a rudimentary form of sentence reduction, as it attempts to give the meaning of multiple words into one “intent”.

The third part of the script deals with output transformation. This, again, is a simplified version of pronoun reversal such as “my -> your” and “you is -> you are”.

The final part of the script deals with keyword transformation, which is effectively the picking out of keywords to attempt to create a cohesive answer. For example, if the script read:  
K I LIKE [string]ING  
R HAVE YOU [string]ED AT ALL RECENTLY?

(With K and R being keyword and response rule starts for the Elizabeth interpreter)  
The user would be able to enter “I like gaming” and the response would be “Have you gamed at all recently?”.

Elizabeth can get very complex, including implementing grammatical rules and advancing its pattern matching algorithm, but the fact remains that Elizabeth is unable to use any sort of “big data” to automatically train itself without assistance from a third-party application, nor can it stray particularly far from exact pattern matches without extremely advanced programming.

The paper ends with a list of problems encountered when dealing with their chosen text database, the Dialog Diversity Corpus, noting that there are no standards formats to distinguish between speakers, sub-standard annotation, and scanned images not being converted into text correctly.

##Citation

Shawar, B. A., & Atwell, E. (2003). Using dialogue corpora to train a chatbot.

- Title: Implementation of ALICE chatbot as domain specific knowledge bot for BRAC U (FAQ bot)  
- Authors: Rahman, Johan  
- Notes: This was a thesis and I sure as hell hope mine isn’t going to be this simplistic.

##Review

A paper[10] by a BRAC University student follows the implementation of a chat bot with domain-specific knowledge about a university, essentially operating as a frequently asked question bot. Their goal is to show how accurate such a system can be, and how it can be improved based on a specific domain.

Domain-specific knowledge is a knowledge base relating to a particular system. In this case, frequently asked questions about BRAC University in Bangladesh. This information was formatted with a modified conversational base already implemented into ALICE, allowing the bot to have its conversation focused only on topics related to the domain.

Creating the bots knowledge bases consisted of extensive brainstorming and writing down as many questions in as many different formats or layouts as possible, to assist in the bot being able to intelligently match patterns, and to make it less likely that a relevant question would be misunderstood or ignored entirely. This included the use of wildcards, for example, “\* about cse370” would assume you were talking about a course, cse370, and would ignore other text.

The bot follows an interesting system for getting an idea about a topic. A user can advance into another topic by mentioning a certain thing, such as “admission”, and the bot will then move on to reading data from the “admission” AIML file instead of the “general” AIML file.

The paper compares between two potential bot setups. The first setup includes only very basic general responses, but the full range of FAQ responses. The second setup includes all general responses from the ALICE source, and the full range of FAQ responses. The first setup intentionally limits conversation from straying too far from the knowledge domain, but potentially could not seem as real or be as comfortable of a conversation for the end user. It was found that in fact the limited setup was preferred, as it came up with much more satisfactory answers a higher percentage of the time, as questions were more to do with domain specific knowledge.

The paper does not explore outside of the ALICE chatbot, nor does it look into any other ways of increasing the size of the knowledge base aside from manually writing new AIML.

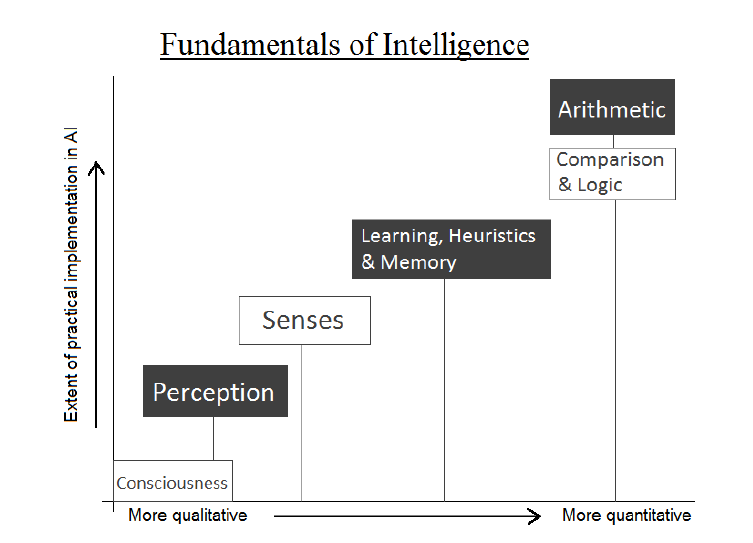
##Citation

Rahman, J. (2012). Implementation of ALICE chatbot as domain specific knowledge bot for BRAC U (FAQ bot).

- Title: A Study of Today’s AI through Chatbots and Rediscovery of Machine Intelligence  
- Authors: Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Pradeepkumar, B., & Das, T  
- Notes:

##Review

According to a paper on today’s AI[11], the evolution of AI has been limited around some key points, such as databases being fed-in manually, or targeting beating the Turing test. Today’s AI systems only “pretend” to act like intelligent entities instead of actually being one. Rule based systems just follow a large set of if-else based logic, rather than applying actual reasoning.

To experiment, they created two chatbot programs. One using traditional AIML, which also gave them the advantage of also using ALICE’s AIML set directly, and one they created in C++ called FUTURE. FUTURE had an advantage over AIML – Being written in C++, it could reference other functions and programs when required, including offering basic arithmetic, trigonometric functions and even differentiating simple expressions.

It was stated that making the AIML bot was significantly easier and faster due to it already having many AI-oriented features. The AIML bot was tested for about 1500 queries, and gave suitable replies to around 1200, for an accuracy of 80%. However, the FUTURE bot was not tested in a similar fashion.

The end goal of the paper was to redefine “Machine Intelligence”. An “Intelligent” system must have all the fundamentals of intelligence (Figure to the right). A partially intelligent system could exhibit some of the fundamentals, such as Chat Bots, which can compare, have logic & reasoning, heuristics, and memory. It can even include the ability to perform arithmetic operations.

##Citation

Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Pradeepkumar, B., & Das, T. (2015). A Study of Today’s A.I. through Chatbots and Rediscovery of Machine Intelligence. *International Journal Science and Technology*, *8*(7), 277–284. https://doi.org/10.14257/ijunesst.2015.8.7.28

- Title: ICE: Enabling Non-Experts to Build Models Interactively for Large-Scale Lopsided Problems  
- Notes:

##Review

Human teachers guiding learning machines presents numerous benefits and challenges according to research by Microsoft[12]. If a teacher can provide additional information and metadata as a learning task progresses, it becomes an *interactive learning* problem. The paper explains the need and benefits of ICE, a platform that accommodates the two-way communication channel needed for efficient interactive learning. The platform has two goals, two produce valuable & deployable models, and to support research on both learning and user interface challenges of the interactive learning problem.

A “lopsided” problem is defined as a problem where not all the data is available. For example, a book review not being correctly labelled as a book review. It could take hundreds of thousands, if not millions of pages for learning machine to correctly predict what a book review is, which is incredibly inefficient. To overcome the issue, they turned to interactive machine learning, which allows human input, training, scoring, and machine feedback in a real-time loop.

Using this method, a single teacher performs all the functions of a domain expert, the labeller and the machine learning expert. At each step, the teacher learns from the machine how the machine is functioning, and machine benefits from the human guidance. The feedback from the machine allows the teacher to gain expertise to best guide the training process without possessing machine learning expertise.

The paper goes on to describe the using of the ICE program, and how your average user would update labels and system states. The paper also describes system architectures, including the extremely powerful servers needed for sub-second delays for the teacher.

While the paper goes into detail on the inner-workings of their own system, they do not really compare to other systems that exist, even though they mention several, such as Information Retrieval. However, their argument is that ICE differs so greatly from other work that it would be erroneous to compare them, as it can be applied to effectively any data type.

##Citation

Simard, P., Chickering, D., Lakshmiratan, A., Charles, D., Bottou, L., Garcia, C., … Suh, J. (2014). ICE: Enabling Non-Experts to Build Models Interactively for Large-Scale Lopsided Problems.

- Title: Fast and easy language understanding for dialog systems with Microsoft Language Understanding Intelligent Service (LUIS)  
- Notes: Short whitepaper

##Review

The Language Understanding Intelligent Service (LUIS) by Microsoft[13] is an attempt at allowing the easier creation of bot dialog systems for developers without machine learning expertise or experience. Being entirely cloud based, developers can set up their model and deploy straight to a HTTP endpoint, where almost anything can interact with its API.

LUIS operates on the idea of “intents”, where the text inputted is compared against various possible programmed “intents” of the user. For example, if you were to enter “start tracking a run”, the possible output from the API may be that the “Start activity” intent with a score of 99%, and the “stop activity” intent with a score of 1%. This meaning that the service is 99% certain that the user wishes to start a new activity, with the entity type of “run”. This information can then be parsed by whatever program is interacting with the API.

The paper is slightly out of date, stating that the service is still in invitation-only beta, though now it is in open beta, but it still has several seemingly up to date images and a relatively descriptive tutorial of how to use the service. However, it does not show any interactions with other software to see how that information can be used or what problems could be faced.

##Citation

Williams, J. D., Kamal, E., Ashour, M., Amr, H., Miller, J., & Zweig, G. (2015). Fast and easy language understanding for dialog systems with Microsoft Language Understanding Intelligent Service (LUIS).

- Title: Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements  
- Notes:

##Review

The Internet of Things consists of a massive number of devices, ranging from sensors, to motors, to communication devices and more, and all of these devices report information to or can be controlled by services. According to *Applying Chatbots to the Internet of Things*[14]there is no reason why these can’t all be controlled in a natural way through chatbots.

The paper defines chatbots as a form of software agent(SA), based on the following key properties that have been associated with SA’s:



Figure 1 - Simple 'Chatbot' example from [14]

* Reactive
* Proactive & Goal Oriented
* Deliberative
* Continual
* Adaptive
* Communicative
* Mobile

The paper quickly makes the connection between chatbots and IoT concerning their use of RESTful Web API’s, citing this as an advantage as developers can take an API or service-oriented approach to development for both IoT and chatbots. Chatbots applications can be deployed side by side on cloud platforms with IoT applications, allowing them to easily communicate with each other and the outside world without worrying about the underlying technologies such as storage and processing.

Several use cases are presented, with the user asking a question and the chatbot giving an example answer. “How much is my car charged” could reply with “The Tesla Model S is currently 40% charged. 3 hours 10 minutes to full charge.”. The Model S already has a smartphone app that allows you to see the current state of the battery, and it would likely not be difficult to view this API from a different source.

Another use case in the image to the right shows a continuous dialog with intent. The user first asks what the status of the dining room is. The bot reports back temperature and lighting conditions. The user continues the conversation without exiting that particular conversation path, so the bot is aware that the user is continuing and applies the command accordingly. With the current advances in smart home automation such as the NEST thermostat, developers could easily hook into the NEST’s API to make this kind of activity possible.

##Citation

Kar, R., & Haldar, R. (2016). Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements.

4 more on network administration/remote network admin/security/etc., then make into one cohesive piece of text that isn’t shit

[1] and M. I. B. Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, “An Efficient Network Monitoring and Management System,” *Int. J. Inf. Electron. Eng.*, vol. 3, no. 1, p. 122, 2013.

[2] R. Collobert, J. Weston, and L. Bottou, “Natural language processing (almost) from scratch,” *J. Mach. …*, vol. 12, pp. 2493–2537, 2011.

[3] N. Chandrika, “Cyber Security in the UK,” *POSTnote*, no. 389, pp. 1–4, 2011.

[4] G. A. Fink, C. L. North, A. Endert, and S. Rose, “Visualizing cyber security: Usable workspaces,” in *6th International Workshop on Visualization for Cyber Security 2009, VizSec 2009 - Proceedings*, 2009, pp. 45–56.

[5] R. Sharma, “Study of Latest Emerging Trends on Cyber Security and its challenges to Society,” *Int. J. Sci. Eng. Res.*, vol. 3, no. 6, 2012.

[6] H. Shiravi, A. Shiravi, and A. a. Ghorbani, “006 A survey of visualization systems for network security,” *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 8, pp. 1313–1329, 2012.

[7] P. M. Nadkarni, L. Ohno-Machado, W. W. Chapman, C. Manning, P. Raghavan, H. Schuetze, W. Hutchins, N. Chomsky, A. Aho, R. Sethi, J. Ullman, N. Chomsky, J. Friedl, S. Kleene, B. Kernighan, R. Pike, J. Levine, T. Mason, D. Brown, A. Joshi, K. Vijay-Shanker, D. Weir, W. Clocksin, C. Mellish, D. Warren, D. Klein, N. Chomsky, D. Klein, C. Manning, J. Quinlan, D. Klein, C. Manning, U. of Pennsylvania, C. Manning, H. Schuetze, D. Jurafsky, J. Martin, P. Spyns, L. Deleger, F. Namer, P. Zweigenbaum, J. Denny, A. Spickard, K. Johnson, S. Haas, G. Savova, J. Masanz, P. Ogren, A. Aronson, Q. Zou, W. Chu, C. Morioka, H. Liu, A. Aronson, C. Friedman, T. Rindflesch, A. Aronson, T. Pedersen, M. Weeber, J. Mork, A. Aronson, W. Chapman, W. Bridewell, P. Hanbury, P. Mutalik, A. Deshpande, P. Nadkarni, Y. Huang, H. Lowe, W. Chapman, W. Bridewell, P. Hanbury, U. of S. (Hungary), G. Savova, W. Chapman, J. Zheng, C. Tao, H. Solbrig, S. Deepak, G. Hripcsak, N. Elhadad, Y. Chen, R. Taira, D. Johnson, V. Bhushan, N. Sager, M. Lyman, N. Nhan, P. Haug, D. Ranum, P. Frederick, L. Christensen, P. Haug, M. Fiszman, H. Xu, C. Friedman, P. Stetson, L. Christensen, H. Harkema, J. Irwin, O. Uzuner, B. South, S. Shen, O. Uzuner, I. Goldstein, Y. Luo, O. Uzuner, O. Uzuner, I. Solti, E. Cadag, C. Chute, L. Chen, C. Friedman, X. Wang, G. Hripcsak, C. Friedman, W. Chapman, M. Fiszman, J. Dowling, W. Chapman, J. Dowling, M. Wagner, X. Wang, G. Hripcsak, M. Markatou, D. Lindberg, B. Humphreys, A. McCray, A. Browne, A. McCray, S. Srinivasan, G. Divita, A. Browne, T. Rindflesch, N. L. of Medicine, S. Srihari, C. Elkan, M. Hearst, S. Dumais, E. Osman, D. Inc., D. Fradkin, I. Muchnik, A. Viterbi, A. Dempster, N. Laird, D. Rubin, M. Hasegawa-Johnson, J. Zhang, D. Shen, G. Zhou, E. Sonnhammer, S. Eddy, E. Birney, A. Lukashin, M. Borodovsky, S. Eddy, L. Rabiner, J. Lafferty, A. McCallum, F. Pereira, A. Culotta, D. Kulp, A. McCallum, C. Sutton, A. McCallum, A. Franz, T. Brants, U. of S. N. L. Group, A. Foundation, A. Foundation, C. Thompson, M. Califf, R. Mooney, N. A. C. of the A. for C. Linguistics, P. Fodor, A. Lally, D. Ferrucci, A. Lally, P. Fodor, L. Libresco, K. Jennings, J. Fitzgerald, E. Berner, G. Webster, A. Shugerman, C. Friedman, A. Elstein, F. Wolf, R. Miller, F. Masarie, and R. Miller, “Natural language processing: an introduction.,” *J. Am. Med. Inform. Assoc.*, vol. 18, no. 5, pp. 544–51, 2011.

[8] R. S. Wallace, “The Anatomy of A.L.I.C.E.” [Online]. Available: http://www.alicebot.org/anatomy.html.

[9] B. A. Shawar and E. Atwell, “Chatbots: are they really useful?,” *LDV-Forum*, 2007.

[10] J. Rahman, “Implementation of ALICE chatbot as domain specific knowledge bot for BRAC U (FAQ bot),” 2012.

[11] A. Khanna, B. Pandey, K. Vashishta, K. Kalia, B. Pradeepkumar, and T. Das, “A Study of Today’s A.I. through Chatbots and Rediscovery of Machine Intelligence,” *Int. J. Sci. Technol.*, vol. 8, no. 7, pp. 277–284, 2015.

[12] P. Simard, D. Chickering, A. Lakshmiratan, D. Charles, L. Bottou, C. Garcia, J. Suarez, D. Grangier, S. Amershi, J. Verwey, and J. Suh, “ICE: Enabling Non-Experts to Build Models Interactively for Large-Scale Lopsided Problems,” 2014.

[13] J. D. Williams, E. Kamal, M. Ashour, H. Amr, J. Miller, and G. Zweig, “Fast and easy language understanding for dialog systems with Microsoft Language Understanding Intelligent Service (LUIS),” 2015.

[14] R. Kar and R. Haldar, “Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements,” 2016.