ASKNet: Creating and Evaluating Large Scale Integrated Semantic Networks

Brian Harrington and Stephen Clark
Oxford University Computing Laboratory
Wolfson Building, Parks Road
Oxford, United Kingdom
{brian.harrington,stephen.clark}@comlab.ox.ac.uk

Abstract

Extracting semantic information from multiple natural language sources and combining that information into a single unified resource is an important and fundamental goal for natural language processing. Large scale resources of this kind can be useful for a wide variety of tasks including question answering, word sense disambiguation and knowledge discovery. A single resource representing the information in multiple documents can provide significantly more semantic information than is available from the documents considered independently. In this paper we describe the ASKNet system, which extracts semantic information from a large number of English texts, and combines that information into a large scale semantic network using spreading activation based techniques. Evaluation of large-scale semantic networks is a difficult problem. In order to evaluate ASKNet we have developed a novel evaluation metric and applied it to networks created from randomly chosen DUC *articles. The results are highly promising: almost 80% precision for the semantic core of the networks.*

1. Introduction

Natural language texts such as newspaper articles and web pages represent a potential gold mine of semantic information. However, in order to realise the potential of this information, we must first be able to extract it from multiple sources and integrate it into a single unified resource. Building large scale semantic resources from multiple natural language texts requires efficient and robust NLP tools, as well as a method for combining the output of those tools in a semantically meaningful way.

The ASKNet system uses NLP tools to extract semantic information from text, and then, through a novel use of spreading activation theory, combines that information into an integrated large scale semantic network. By mapping together concepts and objects that relate to the same real-world entities, ASKNet is able to transform the output of various NLP tools into a single network, producing semantic resources which are more than the sum of their parts. Combining information from multiple sources results in a representation which would not have been possible to obtain from analysing the original sources separately.

The potential of large scale semantic knowledge networks can be seen by the number of projects currently underway to manually construct similar resources. Projects such as Concept Net [13] and Cyc [12] have spent decades of time and thousands of man-hours manually constructing semantic knowledge resources. However, manual construction severely limits the coverage and scale that can be achieved. After more than a decade of work, the largest semantic networks have on the order of 1.5-2.5 million relations connecting 200,000-300,000 nodes [15]. It is necessary therefore to develop systems which can create large scale integrated resources automatically, and produce resources of high quality.

Evaluating semantic networks, especially on the scale of those created by ASKNet, is a difficult problem. We have developed a novel technique for evaluating the semantic "core" of the network. Human evaluators were used to measure the precision of the core for five networks created from randomly chosen DUC articles, resulting in an average accuracy of almost 80%. This highly promising result demonstrates that NLP technology can now be used to create accurate and complex semantic representations of unrestricted text on a very large scale.

2 Extracting Semantic Information

In order to create large scale, integrated semantic networks, ASKNet needs to extract semantic information from text accurately and efficiently. It is only recently that NLP tools capable of achieving this task have become available. ASKNet uses the C&C parser [3], which is is based on the linguistic formalism Combinatory Categorial Grammar (CCG) [16]. CCG is a *lexicalised* grammar formalism, which means that it associates with each word in a sentence an elementary syntactic structure. In CCG's case, these structures are *lexical categories* which encode subcatgeorisation information.

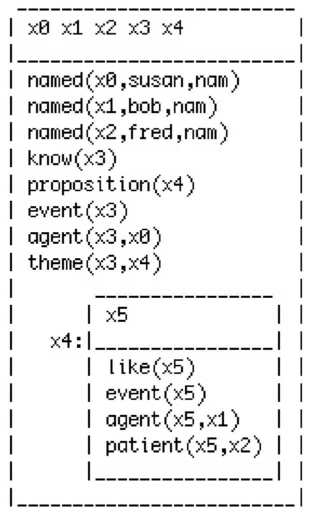
The innovation in the CCG parser is to combine a linguistically-motivated grammar formalism with an efficient and robust parser. The robustness arises from the fact that the grammar is extracted from CCGbank [10], a CCG treebank derived from the Penn Treebank [14], and the use of statistical parsing models trained on CCGbank. CCGbank is based on real-world text: 40,000 sentences of Wall Street Journal text manually annotated with CCG derivations. The efficiency comes from the fact that the lexical categories can be assigned to words accurately using finite-state tagging techniques, which removes much of the practical complexity from the parsing [3].

The C&C parser is part of the C&C NLP toolkit, which also contains a named entity recogniser. A standard approach to named entity recognition is to treat the task as a sequence labelling problem, in which tags are assigned to words in a sentence indicating whether the word is part of a named entity and the entity type. An advantage of this approach is that sequence tagging is a well-understood problem, for which many approaches have been investigated. The C&C NER tagger uses a Maximum Entropy tagger, in which local log-linear models are used to define a distribution over the possible tags, based on the context and the previous two tags. Standard Viterbi decoding can be used to find the most probable sequence of tags. The advantage of using a Maximum Entropy tagger is that it allows great flexibility in terms of the contextual features that can be used to decide on the correct tag; [5] describes the large and varied feature set used by the NER tagger.

The tagger can be trained on any available NER data. In this paper we have used the data from the Message Understanding Conference (MUC), which contains the following semantic categories: `person`, `organisation`, `date`, `time`, `location` and `monetary amount`. The accuracy of the NER tagger ranges from roughly 85 to 90%, depending on the data set and the entity type [5].

Once the data has been parsed, ASKNet uses the semantic analysis tool Boxer [1] to convert the parsed output into a series of first order logic predicates. Boxer has been specifically designed to interpret a CCG derivation and produce a first-order representation, a task which is facilitated by CCG's transparent interface between the syntactic and semantic levels of representation [16]. The semantic theory used is Discourse Representation Theory (DRT) [11].

The output of Boxer is a Prolog style discourse representation structure with variables assigned to objects and first order predicates representing relations between those objects. Boxer captures the underlying semantic relations in a sentence such as "agent" and "patient" to construct labelled and directed relations. Propositions are assigned their own recursively defined sub-structures. Figure 1 gives an example structure, using the standard DRT box-like notation. A simple, low-coverage pronoun resolution scheme is also implemented which attempts to assign appropriate object variables to pronouns. ASKNet can efficiently translate Boxer's semantic output for each sentence into one or more semantic network fragments.



```
 _________________________
| x0 x1 x2 x3 x4          |
|_________________________|
| named(x0,susan,nam)     |
| named(x1,bob,nam)       |
| named(x2,fred,nam)      |
| know(x3)                |
| proposition(x4)         |
| event(x3)               |
| agent(x3,x0)            |
| theme(x3,x4)            |
|        _______________  |
|       | x5            | |
|   x4:|_______________| |
|       | like(x5)      | |
|       | event(x5)     | |
|       | agent(x5,x1)  | |
|       | patient(x5,x2)| |
|       |_______________| |
|_________________________|
```

Figure 1. Example Boxer output for the sentence "Susan knows that Bob likes Fred"

3 Building an Integrated Semantic Network

The semantic networks created by ASKNet consist of object nodes linked by directed labelled relations. The objects and relations roughly correspond to the entity variables and first order relations created by Boxer. In particular, this means that the relations are not bound to a particular set of types, and can be given any label appearing in the Boxer output. This vastly increases the expressiveness of the network.

Another important feature of the network is its nesting structure. ASKNet allows nodes and relations to be combined to form complex nodes which can represent larger and more abstract concepts. These complex nodes can be combined with further relations to represent even more complex concepts. An example is given in Figure 2. The nested structure of the network allows for the expression of complex concepts without having to resort to a rigidly defined structure such as the hierarchical structure used by WordNet [7]. While a pre-defined structure provides a simple and effective framework for network creation, it also limits which nodes may be linked, thereby decreasing the expressiveness of the network.

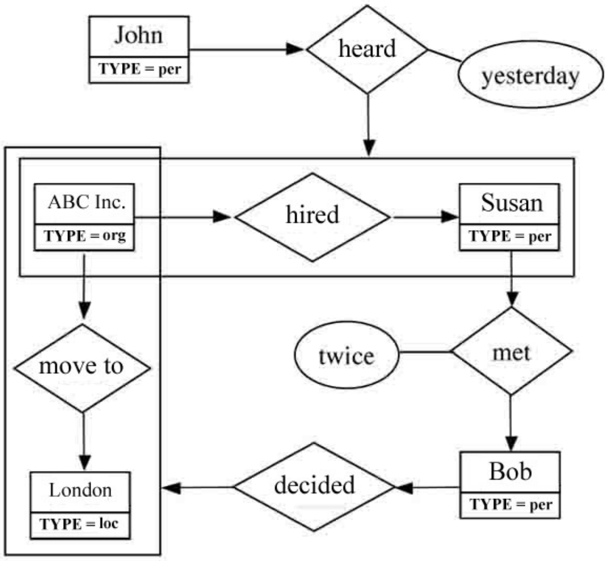


Figure 2. A simplified semantic network created from the sentences "Yesterday John heard that ABC Inc. hired Susan. Bob decided that ABC Inc. will move to London. Susan met Bob twice."

3.1 The Update Algorithm

In order to create a unified network, ASKNet maps nodes in the semantic network fragments which refer to the same real-world entity or concept. This step, performed by the update algorithm, provides a great deal of the potential power of the network, converting a series of small network fragments into a single large-scale semantic knowledge network.

The update algorithm uses spreading activation theory [4] to determine which nodes are co-referent. When a smaller *update* network is combined with the larger knowledge network, some of the nodes in the update network may refer to the same real world entities as existing nodes in the knowledge network. Potential node pair matches are initially scored based on lexical information, and then spreading activation is used to gradually refine the scores. Scores above a certain threshold indicate that the two nodes refer to the same real world entity and should be mapped together.

In order to understand the operation of the update algorithm, we will walk through a single iteration of a simplified example shown in Figure 3. All nodes will be referred to by their node ID; thus go refers to the node with the label Gore in the update network, while algore refers to the node with the label Al Gore in the main knowledge network. The technical details of the update algorithm, and a more detailed version of this example are covered in [9].

When trying to integrate the networks, the update algorithm first pairs each update network node with each main network node, and assigns an association score to each node pair according to lexical and named entity type similarity. In this instance the algorithm will give the same association score to (bu,johnbush) as it would to (bu,georgebush), as they have the same lexical and named entity type similarity.

The algorithm first attempts to improve the score for the bu node pairings. After adding some activation to the bu node, and firing the update network, the go and wh nodes will receive an amount of activation dependent on the length and strength of their links with bu. The update algorithm then transfers this activation to the main network, with the activation from wh going to whitehouse, and the activation from go being split between gorevidal and algore according to their current association scores.

Firing the main network will now cause the activation from whitehouse and algore to move through the network, with some of the activation reaching the georgebush node. Since the georgebush node receives activation, and the johnbush node receives none, we have an increased confidence that the pairing (bu,georgebush) is correct, and thus the update algorithm increases the association score for this pairing accordingly, while simultaneously decreasing the score for (bu,johnbush).

In future iterations, the activation from bu will be transferred in greater proportion to georgebush, thus causing a mutually reinforcing loop that will eventually result in the association scores for (bu,georgebush),(go,algore) and (wh,whitehouse) increasing with each iteration, and the scores for (bu,johnbush) and (go,gorevidal) decreasing to zero.

The update algorithm continues in this way until a set number of iterations have been completed, or the change in association scores between iterations drops below a given value. Node pairs with an association score above a set threshold are then mapped together, and the update network is integrated into the main knowledge network.

The use of spreading activation to combine network fragments from multiple sources greatly increases the amount of semantic information which can be obtained from the network. Consider the example illustrated in Figure 4. By interpreting the network fragments alone, it would not be possible to find the potential link between *Disease B* and *Chemical F*. Combining the fragments into a single integrated network produces a resource containing semantic information not readily available in the documents themselves.

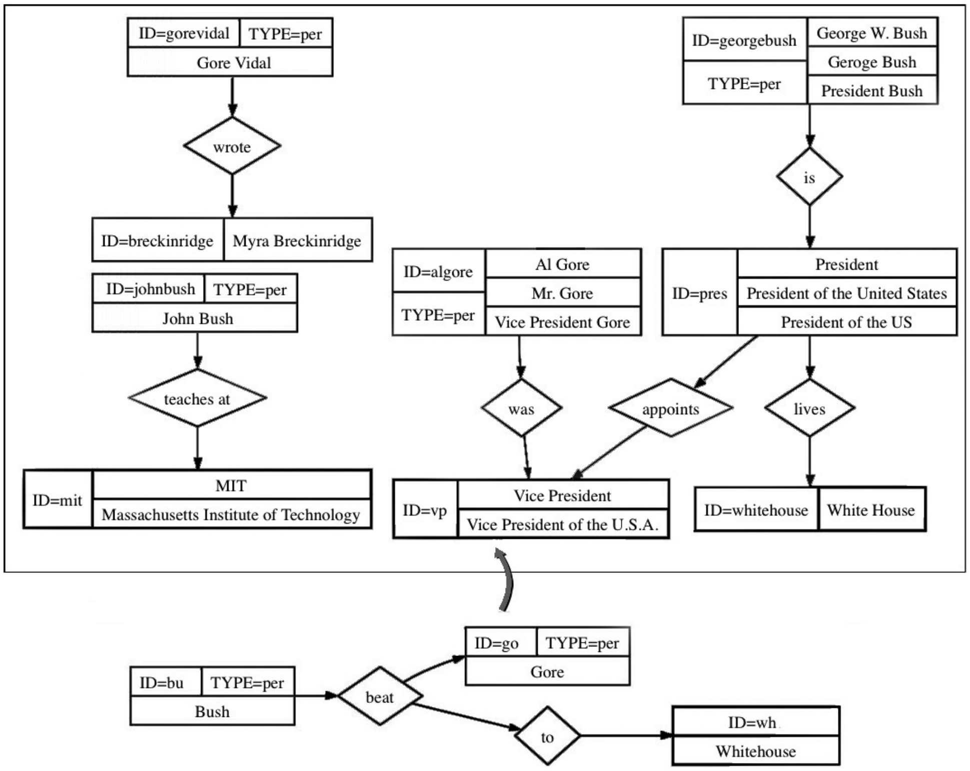


Figure 3. An example update network created from the sentence "Bush beat Gore to the Whitehouse" being added to a network containing information about United States politics, writers and mathematicians.

4 Evaluation

4.1 Network Creation Speed

By processing approximately 2 million sentences of newspaper text from the New York Times, we were able to build a network of over 1.5 million nodes and 3.5 million links in less than 3 days. This time also takes into account the parsing and semantic analysis (See Table 1). This is a vast improvement over manual approaches which take years or even decades to build networks of less than half this size [15].

As the network grows, the time to perform the information integration step begins to climb exponentially. However, because the spreading activation algorithms are localised, once the network becomes so large that the activation does not spread to the majority of nodes, any increase in size ceases to have an effect on the algorithm. Therefore the average time to add a new node to the network is asymptotic as seen in Figure 5 and will eventually become constant regardless of network growth.

Total Number of Nodes	1,500,413
Total Number of Edges	3,781,088
Time: Parsing	31hrs : 30 min
Time: Semantic Analysis	16 hrs: 54 min
Time: Building Network & Information Integration	22 hrs : 24 min
Time: Total	70 hrs : 48 min

Table 1. Statistics relating to the creation of a large scale semantic network

4.2 Network Precision

Evaluating large-scale semantic networks is a difficult task. Traditional NLP evaluation metrics such as precision and recall do not apply so readily to semantic networks; the networks are too large to be directly evaluated by humans; and even the notion of what a "correct" network should look like is difficult to capture.

NLP evaluation metrics also typically assume a uniform

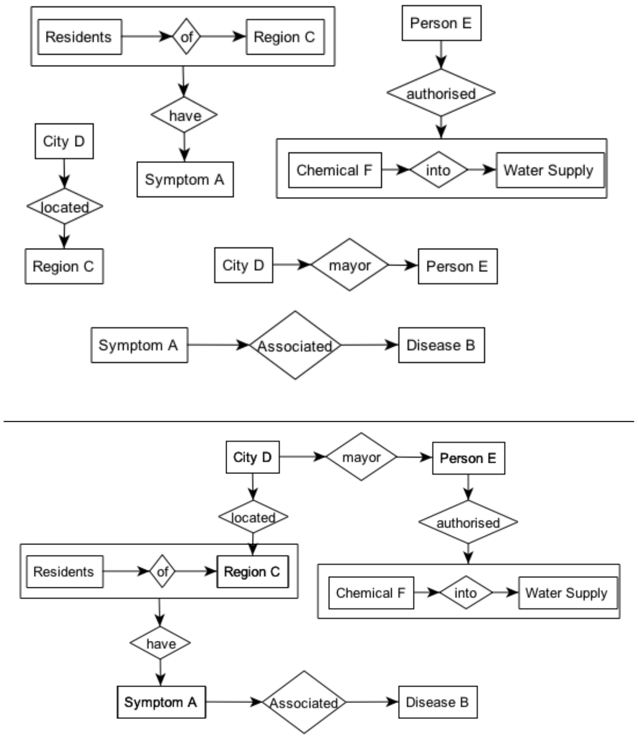


Figure 4. A simple example network taken from multiple source types before and after integration

importance of information. However, when considering semantic networks, there is often a distinction between relevant and irrelevant information. For example, a network containing information about the Second World War could contain the fact that September 3rd 1939 was the day that the Allies declared war on Germany, and also the fact that it was a Sunday. Clearly for many applications the former fact is much more relevant than the latter. In order to achieve a meaningful precision metric for a semantic network, it is important to focus the evaluation on high-relevance portions of the network.

There is no gold-standard resource against which these networks can be evaluated, and given their size and complexity it is highly unlikely that any such resource will be built. Therefore evaluation can either be performed by direct human evaluation or indirect, application based evaluation. For this paper we have chosen direct, human evaluation.

The size of the networks created by ASKNet makes human evaluation of the entire network impossible. It is therefore necessary to define a subset of the network on which to focus evaluation efforts. In preliminary experiments, we found that human evaluators had difficulty in accurately evaluating networks with more than 20–30 object nodes and 30–40 relations.

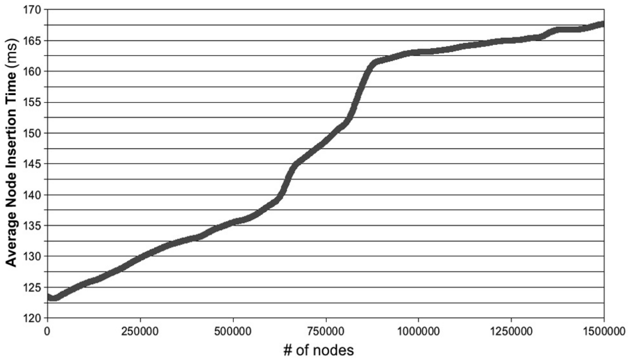


Figure 5. Average time to add a new node to the network vs. total number of nodes

Rather than simply evaluate a random subset of the network, which may be of low-relevance, we evaluated a network "core", which we define as a set of high-relevance nodes, and the network paths which connect them. This allows us to maintain a reasonable sized network for evaluation, while still ensuring that we are focusing our efforts on the high-relevance portions of the network.

We evaluated networks based on documents from the 2006 Document Understanding Conference (DUC). These documents are taken from multiple newspaper sources and grouped by topic. This allows us to evaluate ASKNet on a variety of inputs covering a range of topics, while ensuring that the update algorithm, which deals with coreference resolution, is tested by the repetition of entities across documents. In total we used 125 documents covering 5 topics, where topics were randomly chosen from the 50 topics covered in DUC 2006. The topics chosen were: Israeli West Bank Settlements, Computer Viruses, NASA's Galileo Mission, the 2001 Election of Vladimir Putin and Elian Gonzalez Custody Battle.

4.2.1 Building the Core

Our task in building the core is to reduce the size of the evaluation network while maintaining the most relevant information for this particular type of network (newspaper text).

We begin to build the core by adding all named entity nodes which are mentioned in more than 10% of the documents. In evaluating the DUC data, we find that over 50% of the named entity nodes are only mentioned in a single document (and thus are very unlikely to be central to the understanding of the topic). This reduces the number of named entities to an average of 12 per network while still ensuring that the most important entities remain in the core.

For each of the named entity nodes in the core, we perform a variation of Dijkstra's algorithm [6] to find the strongest path to every other named entity node in the core. Rather than using the link weights to determine the shortest path, as in the normal Dijkstra's algorithm, we use the spreading activation algorithm to determine the path which sends the greatest amount of activation, which we will call the *primary* path. Adding all of these paths to the core results in a representation containing the most important named entities in the network, and the primary path between each pair of nodes (if such a path exists).

The core that results from the Dijkstra-like algorithm focuses on the relationships between the entities and discards peripheral information about individual entities within the network. It also focuses on the strongest paths, which represent the most salient relationships between entities and leaves out the less salient relationships (represented by the weaker paths).

Running this algorithm on the networks produced from the DUC data results in cores with an average of 20 object nodes and 32 relations per network, which falls within the acceptable limit for human evaluation. An additional benefit of building the core in this manner is that the resulting core tends to contain the most salient nodes and relations in the network, and thus allows human evaluators to easily identify which portions of the network relate to which aspect of the stories.

We also found during our experiments that the core tended to stabilise over time. On average only 2 object nodes and no named entity nodes changed within the core of each network between inputting the 20th and the 25th document of a particular DUC category. This indicates that the core, defined in this way, is a relatively stable subset of the network, and represents information which is central to the story, and is therefore being repeated in each article.

4.2.2 Evaluating the Core

ASKNet uses the GraphViz [8] library to produce graphical output. This allows human evaluators to quickly and intuitively assess the correctness of portions of the network. One network was created for each of the 5 topics, and graphical representations were output for each network. An example of the graphical representation of the network cores used for evaluation is shown in Figure 6. The representation is similar to that in Figure 2, with nodes (rectangles) representing entities, and links (diamonds) representing relations. To ease the evaluator's task, we have chosen to output the graphs without the recursive nesting. In some cases, connector nodes (ovals) were added to provide information that was lost due to the removal of the nesting.

Each of the 5 topic networks was evaluated by 3 human evaluators. (The networks were distributed in such a way as to ensure that no two networks were evaluated by the same 3 evaluators). The evaluators were provided with the graphical output of the networks they were to assess, the sentences that were used in the formation of each path, and a document explaining the nature of the project, the formalities of the graphical representation, and the step-by-step instructions for performing the evaluation.[1]

The evaluation was divided into 2 sections and errors were classified into 3 types. The evaluators were first asked to evaluate the named entity nodes in the network, to determine if each node had a *type error* (an incorrect named entity type as assigned by the named entity tagger e.g., a node referring to a person having type *org*), or a *label error* (an incorrect set of labels, indicating that the node did not correspond to a single real world entity e.g., labels from multiple entities being added to the same node). The evaluators were then asked to evaluate each primary path. If there was an error at any point in the path (e.g., a relation attached to the wrong node), the entire path was deemed to be incorrect.

The error types were divided in an attempt to discover their source. Type errors are caused by the named entity tagger, label errors by the update algorithm or the semantic analyser (Boxer), and path errors by the parser or Boxer.

4.2.3 Evaluation Results

The scores reported by the human evaluators are given in Table 2. The scores given are the percentage of nodes and paths that were represented entirely correctly. A named entity node with either a type or label error was considered incorrect, and any path segment containing a path error resulted in the entire path being labelled as incorrect. The overall average precision was 79.1%, with a Kappa Coefficient [2] of 0.69 indicating a high level of agreement between evaluators.

Topic	Eval 1	Eval 2	Eval 3	Avg
Elian Gonzalez	88.2%	70.1%	75.0%	77.6%
Galileo Probe	82.6%	87.0%	91.3%	87.0%
Viruses	68.4%	73.7%	73.7%	71.9%
Vladimir Putin	90.3%	82.8%	94.7%	89.9%
West Bank	68.2%	77.3%	70.0%	72.3%
Average Precision:				79.1%

Table 2. Evaluation Results

Due to the nature of the evaluation, we can perform further analysis on the errors reported by the evaluators, and categorize each error by type as seen in Table 3. The results in Table 3 indicate that the errors within the network are not from a single source, but rather are scattered across each of the steps. The *NE Type* errors were made by the

[1] All of the evaluation materials provided to the evaluators can be downloaded at: `http://www.brianharrington.net/asknet`.

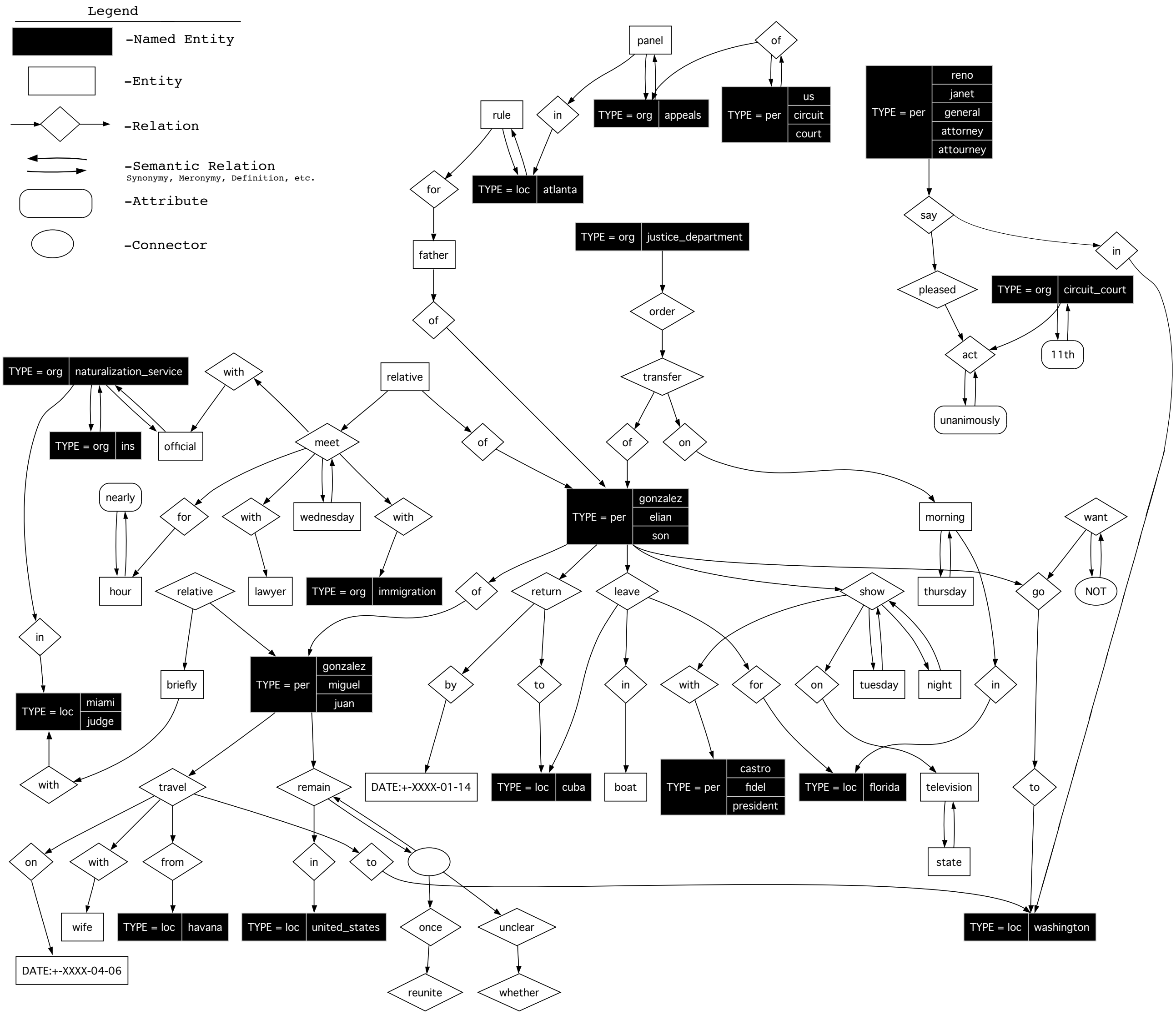


Figure 6. Graphical representation for topic: "Elian Gonzalez Custody Battle".

Topic	NE Type	Label	Path
Elian Gonzalez	8.3%	50.5%	41.7%
Galileo Probe	22.2%	55.6%	22.2%
Viruses	93.8%	0.0%	6.3%
Vladimir Putin	22.2%	33.3%	44.4%
West Bank	66.7%	27.8%	5.6%
Total:	43.4%	32.9%	23.7%

Table 3. Errors by Type

NER tool. The *Label* errors came from either Boxer (mostly from mis-judged entity variable allocation), or from the Update Algorithm (from merging nodes which were not co-referent). The *Path* errors were caused by either the parser mis-parsing the sentence, Boxer mis-analysing the semantics, or from inappropriate mappings in the Update Algorithm.

The errors appear to be relatively evenly distributed, indicating that, as each of the tools used in the system improves, the overall quality of the network will increase. Some topics tended to cause particular types of problems. Notably, the NER tool performed very poorly on the Viruses topic. This is to be expected as the majority of the named entities were names of computer viruses or software programs that would not have existed at all in the training data.

An overall precision of 79.1% is highly promising for such a difficult task. The high score indicates that, while semantic network creation is by no means a solved problem, it is possible to create a system which combines multiple natural language inputs into a single cohesive knowledge network and does so with a high level of precision. In particular we have shown that ASKNet's use of spreading activation techniques results in a high quality network core, with the most important named entities and the relations between those entities being properly represented in the majority of cases.

5 Future Work

We plan to evaluate the network on a larger scale using a task based evaluation. By using networks created by ASKNet to perform a particular task such as semantic distance calculation or word sense disambiguation, and comparing the results with those of systems designed specfically for that task, we can obtain some indication of how useful these networks can be in particular applications.

The evaluation results obtained in this paper indicated that the primary entities within the network core quickly become well connected. We plan to utilise this fact to develop systems based on connectivity ranking, a technique which would allow ASKNet to determine how closely two entities are related using techniques similar to the existing spreading activation algorithms implemented within the network. This would allow us to compute semantic distance between entities, but could also discover interesting relationships between entities that do not have any direct connection in a single document. This could be especially useful in domains such as biomedicine, where there are many biological entities which may have interesting biological relationships, but whose interactions have never been tested in a laboratory.

6 Conclusion

This paper described the use of the ASKNet system to create large scale semantic resources from multiple natural language documents. We have argued that integrating the information retrieved from individual documents into a single unified resource is a difficult and interesting problem, and that this integration greatly improves the usefulness of the resulting network.

Evaluating semantic networks is a difficult task. In order to evaluate ASKNet, we developed a novel evaluation method based on human evaluators measuring the precision of the network core. Using this metric we obtained results of almost 80% for five ASKNet networks created from randomly chosen DUC articles. This highly promising result shows that it is possible to efficiently construct high quality, coherent semantic resources from multiple natural language documents, using state-of-the-art NLP tools and a novel use of spreading activation.

References

[1] J. Bos, S. Clark, M. Steedman, J. R. Curran, and J. Hockenmaier. Wide-coverage semantic representations from a CCG parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, pages 1240–1246, Geneva, Switzerland, 2004.

[2] J. Carletta. Assessing agreement on classification tasks: the Kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

[3] S. Clark and J. R. Curran. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552, 2007.

[4] A. M. Collins and E. F. Loftus. A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407–428, 1975.

[5] J. R. Curran and S. Clark. Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-03)*, pages 164–167, Edmonton, Canada, 2003.

[6] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerical Mathematics*, 1:269 – 271, 1959.

[7] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[8] E. R. Gansner and S. C. North. An open graph visualization system and its applications to software engineering. *Software — Practice and Experience*, 30(11):1203–1233, 2000.

[9] B. Harrington and S. Clark. ASKNet: Automated semantic knowledge network. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI'07)*, pages 889–894, Vancouver, Canada, 2007.

[10] J. Hockenmaier. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh, 2003.

[11] H. Kamp and U. Reyle. *From Discourse to Logic : Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic, Dordrecht, 1993.

[12] D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33 – 38, 1995.

[13] H. Liu and P. Singh. ConceptNet a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22:211 – 226, Oct 2004.

[14] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1994.

[15] C. Matuszek, J. Cabral, M. Witbrock, and J. DeOliveira. An introduction to the syntax and content of Cyc. In *2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, Stanford, CA, USA, March 2006.

[16] M. Steedman. *The Syntactic Process*. The MIT Press, Cambridge, MA., 2000.