

ASKNET: CREATING AND EVALUATING LARGE SCALE INTEGRATED SEMANTIC NETWORKS

BRIAN HARRINGTON

*Oxford University Computing Laboratory
Wolfson Building, Parks Rd.
Oxford, UNITED KINGDOM
brian.harrington@comlab.ox.ac.uk
<http://www.brianharrington.net/asknet>*

STEPHEN CLARK

*Oxford University Computing Laboratory
Wolfson Building, Parks Rd.
Oxford, UNITED KINGDOM
stephen.clark@comlab.ox.ac.uk*

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.

The ASKNet system utilises existing NLP tools and resources, together with spreading activation based techniques, to automatically extract semantic information from a large number of English texts, and combines that information into a large scale semantic network. The initial emphasis of the ASKNet system is on wide-coverage, robustness and speed of construction. In this paper we show how a network consisting of over 1.5 million nodes and 3.5 million edges, more than twice as large as any network currently available, can be created in less than 3 days. Evaluation of large-scale semantic networks is a difficult problem. In order to evaluate ASKNet we have developed a novel evaluation metric based on the notion of a network “core” and employed human evaluators to determine the precision of various components of that core. We have applied this evaluation to networks created from randomly chosen articles used by DUC (the Document Understanding Conference). The results are highly promising: almost 80% precision in the semantic core of the networks.

Keywords: ASKNet, Semantic Networks, Knowledge Extraction, Semantic Network Evaluation, Wide-coverage Semantic Analysis

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 natural language processing (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 produce a single unified entity relationship style semantic network. Combining information from multiple sources results in a representation which can reveal information that could not have been obtained from analysing the original sources separately.

Systems aiming to build semantic resources of this type have typically either overlooked information integration completely, or else dismissed it as being AI-complete, and thus unachievable. In this paper we will show that information integration is both an integral component of any semantic resource, and achievable through a combination of NLP technologies and novel applications of spreading activation theory. While complete extraction and integration of all knowledge within a text may be AI-complete, we will show that by processing large quantities of text efficiently, we can compensate for minor processing errors and missed relations with volume and creation speed. If relations are too difficult to extract, or we are unsure which nodes should integrate at any given stage, we can simply leave them to be picked up later when we have more information or come across a document which explains the concept more clearly.

ASKNet is capable of efficiently generating integrated semantic networks on a scale never seen before. In this paper we demonstrate the speed and robustness of ASKNet by creating networks consisting of over 1.5 million nodes and 3.5 million edges — more than twice as large as any existing semantic network — in less than 3 days. In the second half of the paper, we explore the problem of evaluating such networks.

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.

This work does not propose to extract all possible semantic information from every input sentence, or to build a complete and perfect semantic representation. Doing so would be an AI-hard task. However, current NLP technologies have reached a stage where extracting basic semantic relations between entities in naturally occurring text has become possible. By utilising these technologies, and integrating the extracted information into a unified resource, ASKNet can efficiently produce large scale, high quality, useful semantic resources.

2. Related Work

The potential of a large scale semantic knowledge base can be seen by the number of projects currently underway to build one. Projects such as Concept Net [1] and Cyc [2] have spent years of time and thousands of man-hours manually constructing semantic knowledge networks. 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 [3].

These networks have been shown to be useful in tasks such as question answering [4] and predictive text entry [5]. However, many tasks either require a domain specific knowledge base, which needs to be created quickly for the task at hand, or require much wider coverage than is possible to achieve in manually created networks. Automatic generation allows us to acquire information from existing data to create new semantic resources very quickly, and to create resources which are many orders of magnitude larger.

One existing system which automatically creates semantic networks is MindNet [6]. MindNet uses a natural language parser to extract pre-defined relations from dictionary definitions. To illustrate the time difference for automated construction over manual creation, the MindNet network of over 150,000 words, connected by over 700,000 relations (roughly half the size of the ConceptNet or Cyc networks), can be created in a matter of hours on a standard personal computer [7]. The difference between the ASKNet system and MindNet is that MindNet is limited to building networks with a small, pre-defined set of relations, and limited to extracting knowledge from well-formed data such as dictionaries. In contrast, ASKNet extracts the relations from the text itself using a wide-coverage parser. ASKNet also integrates information from multiple sources by mapping together nodes which refer to the same real-world entity; a task which is not attempted by MindNet. This allows ASKNet to accommodate a much wider variety of information, use more varied sources of input, and extract more information than any similar system currently in development.

The Espresso system [8] attempts to harvest semantic relations from natural language text by building word patterns which signify a specific relation (e.g., “X consists of Y” for the `part_of(Y,X)` relation) and searching large corpora for text which fits those patterns. The building of patterns is weakly-supervised, and each new relation the system extracts must be chosen by a human user. Unlike ASKNet, Espresso only extracts binary relations, and does not build complex node structures or perform any information integration.

Schubert and Tong ([9]) have also developed a system for automatically acquiring knowledge from text. However, they attempt to gain “possibilistic propositions” (e.g., “*A person can believe a proposition*”) rather than extracting direct knowledge from the text. Furthermore, they only extract information from a small treebank corpus rather than raw text. ASKNet can extract information from raw text because

of its use of a wide coverage parser. This allows us to use the vast quantities of readily available English text to create networks, instead of comparatively small structured corpora.

Semantic resources created automatically will contain more errors than their manually created counterparts. However, for many tasks, the great decrease in time and labour required to build a network, combined with the ability to create extremely large networks, is a worthwhile trade-off for any decrease in accuracy [6].

3. Parsing and Semantic Analysis

Manipulating natural language in its raw form is a difficult task. Parsers and semantic analysis tools allow us to work with the content of a document on a semantic level. This simplifies the process of developing a semantic network from a computational standpoint, and allows us to focus on the higher level semantic tasks, without the requirement of dealing with the lexical and syntactic levels. To this end, ASKNet employs a set of language processing tools to render the plain text into a discourse representation structure, from which point it can turn the information into a semantic network fragment with relative ease.

Creating very large networks with a variety of relation types requires a parser which is robust, efficient, and which has wide-coverage. It is only recently that such parsers have become available. ASKNet uses the C&C parser [10], which is based on the linguistic formalism Combinatory Categorical Grammar (CCG) [11]. 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 subcategorisation 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 [12], a CCG treebank derived from the Penn Treebank. 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 [10].

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. 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. [13] 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 MUC data, 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 [13].

Once the data has been parsed, ASKNet uses the semantic analysis tool Boxer [14] 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 [11]. The semantic theory used is Discourse Representation Theory (DRT) [15]. 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.

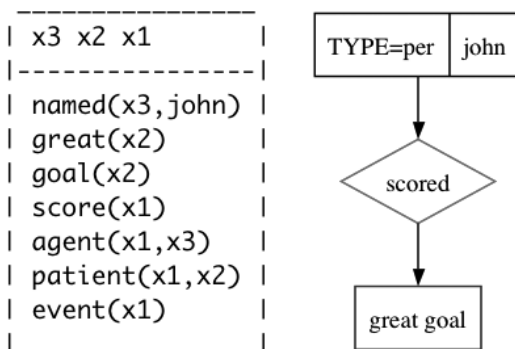


Fig. 1. Example Boxer output and corresponding network fragment for the sentence “John scored a great goal.”

The ASKNet framework has been designed to be flexible, and could easily be adapted to other NLP tools. However, we have chosen to use the Clark and Curran parser and Boxer because of their efficiency, coverage, robustness, and the relative sophistication of their output.

4. The Network

The semantic networks created by ASKNet consist of object nodes and attribute 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 pre-defined set of types, and can be given any label appearing in the Boxer output, which 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 network 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 [16]. 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.

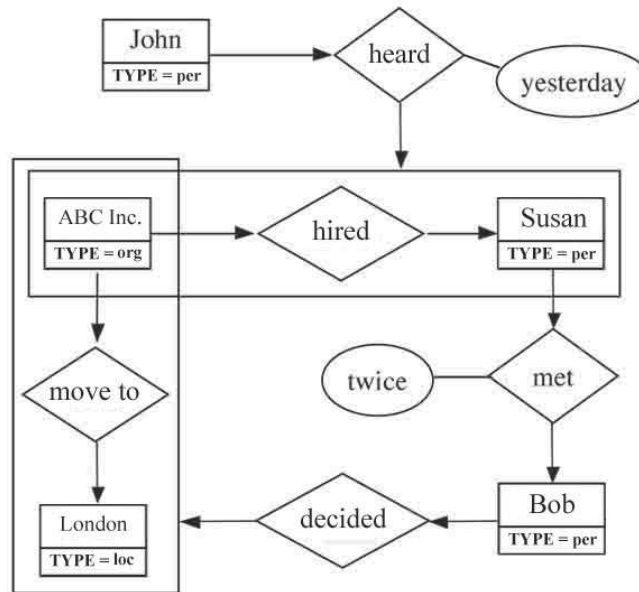


Fig. 2. A simplified Semantic Network created from the sentences “Yesterday John heard that ABC Inc. hired Susan. Bob decided that ABC Inc. will move to London. Susan met Bob twice.”

Each relation in the network has a weight which represents the confidence of the network in the “real world” existence of the relation and also its salience. Factors

such as the confidence of the initial sentence parse, the source of the information, how recently the information has been encountered and the number of different sources verifying the information could all affect the weight given to a relation. For example, the program currently sets the weights of the relations based on the number of times that the relation has been seen, and also increases the weight of relations gathered from headlines over those gathered from the body of a document.

4.1. *Spreading Activation*

Concepts and ideas in the human brain have been shown to be semantically linked [17] so that thinking about (or firing) one concept primes other related concepts making them more likely to fire in the near future. This is the idea behind the ASKNet network. Firing a node sends activation out to all of the nodes semantically connected to that node, which can either cause them to fire (analogous to one concept “triggering” thoughts of a related concept) or, if there is not enough activation to trigger a firing, cause them to store activation, making them more likely to fire in the future (analogous to “priming” in the human brain). The weight of the relations connecting two nodes determines the amount of activation that is transferred after a node fires.

By firing one or more nodes and analysing the way in which activation spreads through the network, we can determine the semantic distance between various entities and concepts. This allows us to determine how closely related two entities or concepts are even if they are not directly linked. When a test node is fired, we can measure how closely related any sample node is to our test node simply by measuring the total activation that came to the sample node through all of its links.

In ASKNet, all nodes maintain an activation level, and a firing threshold. If their activation level exceeds their firing threshold at any time, they will fire, sending an amount of activation to each of their neighbouring nodes based on the activation function given in Eq. (1).

Spreading activation is an efficient means of determining semantic distance in the network because it is localised. Most search based algorithms would require traversal of the entire network before calculating the total degree of connectivity. Spreading activation only accesses a small number of nodes in a localised area, and thus the amount of time required for a single firing depends only on the amount of initial activation provided, and the amount of activation that is lost with each transfer. Hence, as the network grows in size, the time complexity of the firing algorithm remains constant.

5. Information Integration

A key problem when building the network is deciding which nodes are co-referent. Information integration of this sort allows the network to become more connected and provides a great deal of the potential power of the network. Without this step the network would simply be a series of small unconnected network fragments.

$$activation_{i,j} = \frac{weight_{i,j}}{\sum_{k|k \in link(i), k \neq j} \beta_{i,k} weight_{i,k}} \quad (1)$$

Symbol Definitions

| | |
|--------------------|--|
| α_i | Firing variable which fluctuates depending on node types |
| $activation_{x,y}$ | Amount of activation sent from node x to node y when node x fires |
| $weight_{x,y}$ | Strength of link between node x and node y |
| $\beta_{x,y}$ | Signal attenuation on link (x,y), $0 < \beta < 1$ determines the amount of activation that is lost along each link. Fluctuates depending on link types |
| $link(x)$ | The set of nodes y such that link(x,y) exists |
| $link(x, y)$ | The directed link from node x to node y |

However, coreference resolution is a difficult task since it often requires semantic as well as lexical information.

5.1. The Update Algorithm

The update algorithm is the process by which ASKNet merges sentence level networks into document level networks, and merges document level networks into the overall knowledge network. The basic premise behind the algorithm is that 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, updating the knowledge network with the network fragment 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 “A1 Gore” in the knowledge network.

Initially, all potential node pairs are scored based on named entity type and label similarity. All nodes contain a set of labels, extracted directly from the Boxer output. The label similarity score is based on the percentage of labels having an edit distance below a set threshold, with label order being disregarded. The named entity type similarity score is a set value which is added if both nodes have the same assigned named entity type. These scores are entered into the *similarity matrix* given in Table 1.

Table 1 shows that the initial scoring is more likely to match `bu` with `bush` instead of the correct matching with `georgebush`. This is because the labels in `bu`

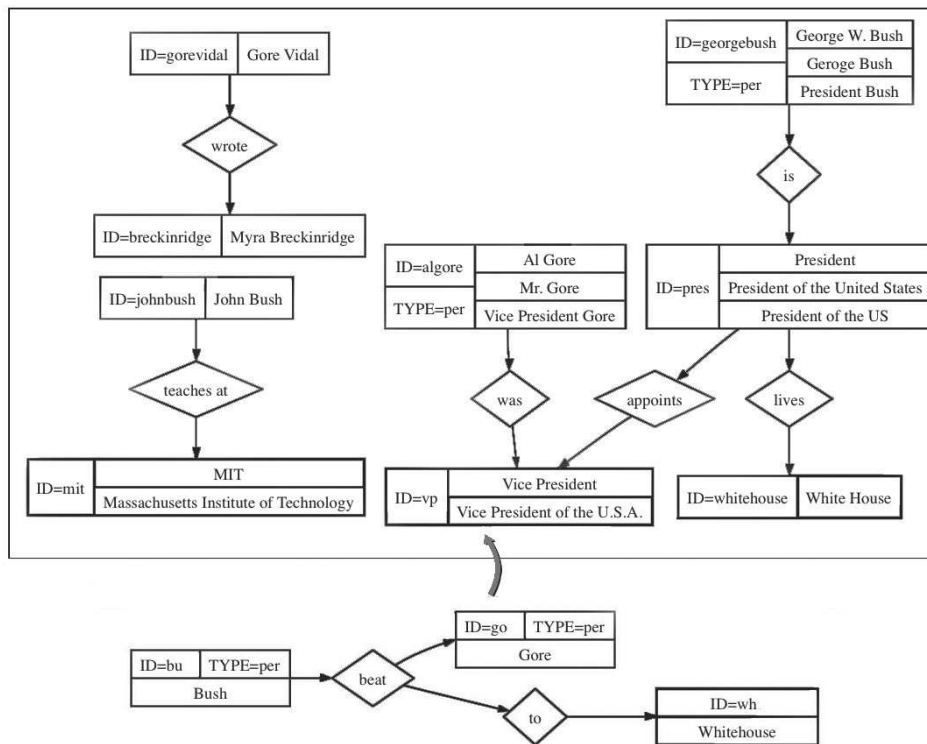


Fig. 3. An example update network created from the sentence “Bush beat Gore to the Whitehouse” being added to a network containing information about United States politics, along with some extraneous information regarding famous authors and university lecturers.

| | georgebush | bush | algore | gore | whitehouse |
|----|------------|------|--------|------|------------|
| bu | 0.5 | 0.7 | | | |
| go | | | 0.5 | 0.7 | |
| wh | | | | | 0.8 |

Table 1. Similarity Matrix: Initial scoring

and **bush** are identical, which outscores the named entity type similarity in **bu** and **georgebush**. Similarly, the initial scoring shows an intial best match of **go** with **gore** instead of the correct match with **algore**.

Once the initial scoring is completed, the algorithm chooses a node to be evaluated from the update network (in this case **bu**) and attempts to improve the scores in its row. The **bu** node is fired in the update network, which sends activation to the **go** and **wh** nodes. For all nodes in the update network which received more than a minimum threshold of activation, their corresponding nodes are fired in the main

network, with an initial activation level determined by their similarity score. For example, the amount of initial activation which the **algore** node receives is based on the activation level of the **go** node and the similarity score between **algore** and **go**.

The **whitehouse** and **gore** nodes will fire in the main network, with the **gore** node receiving slightly more activation than the **algore** node, because of its higher similarity score. This firing pattern will cause activation to spread throughout the network; the **georgebush** node will receive some activation from the firing, while the **bush** node will not receive any.

Since the **georgebush** node received activation, its similarity score with the original evaluation node (**bu**) will be increased, while the similarity score between the **bush** and **bu** nodes will be decreased. Table 2 shows some typical scores resulting from this stage of the process.

| | georgebush | bush | algore | gore | whitehouse |
|----|------------|------|--------|------|------------|
| bu | 0.7 | 0.35 | | | |
| go | | | 0.5 | 0.7 | |
| wh | | | | | 0.8 |

Table 2. Similarity Matrix: After evaluating the **bu** node

The **go** node is then evaluated, with almost identical results, except that the **georgebush** node will fire with more activation than the **bush** node because of its improved score from the previous step, which results in the **algore** node receiving an even stronger score improvement. The combined improved results produce an even stronger effect when the **wh** node is evaluated. After one iteration, the similarity matrix is as given in Table 3.

| | georgebush | bush | algore | gore | whitehouse |
|----|------------|------|--------|------|------------|
| bu | 0.7 | 0.35 | | | |
| go | | | 0.8 | 0.35 | |
| wh | | | | | 0.95 |

Table 3. Similarity Matrix: After one iteration

After several more iterations, the similarity scores between **bu** and **georgebush**, **go** and **algore**, and **wh** and **whitehouse** will increase, and all other scores will drop to zero. Once a stopping criterion (number of iterations, or minimum change in scores) has been met, any node pairs with a similarity score above a pre-set threshold are assumed to denote the same real-world entity and are mapped together, merging their links and labels as well as internal variables, such as activation thresholds and firing history.

In this instance, lexical as well as semantic information was used to determine that the `bu`, `go` and `wh` nodes referred to George Bush, Al Gore and the United States’ White House respectively. This was a simplified example, but the principle can be extended to deal with more complex networks.

6. Evaluation

6.1. 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 4), and is a vast improvement over manually created networks for which years or even decades are required to achieve networks of less than half this size [3].

| | |
|--|-----------------|
| 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 4. Statistics pertaining to the creation of a large scale semantic network

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 4 and will eventually become constant regardless of network growth.

6.2. 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 define.

NLP evaluation metrics also typically assume a uniform 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

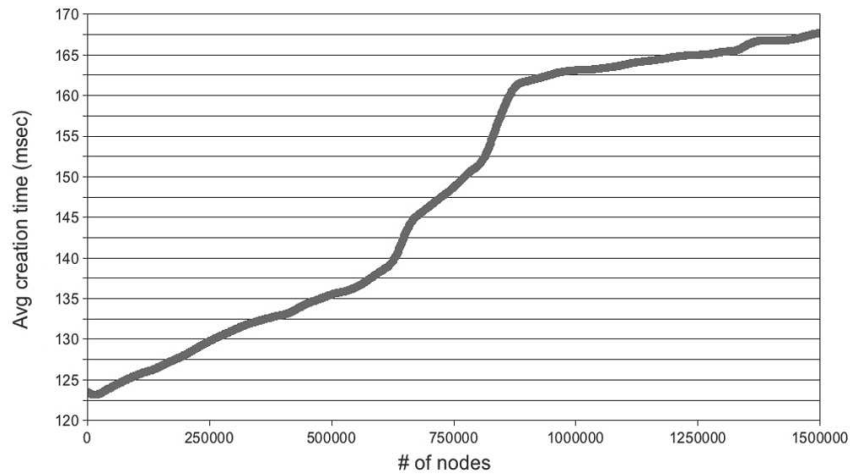


Fig. 4. Average time to add a new node to the network vs. total number of nodes

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 make 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 early experiments, we found that human evaluators had difficulty in accurately evaluating networks with more than 20 – 30 object nodes and 30 – 40 relations. Rather than simply evaluating 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 in the high-relevance portions of the network. These are also likely to be the portions of the network which have undergone the most iterations of the update algorithm. Therefore the evaluation will be more likely to give an accurate representation of ASKNet’s overall capability, rather than being dominated by the quality of the NLP tools used.

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 the Elian Gonzalez Custody Battle.

6.2.1. 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 (a value picked for pragmatic purposes of obtaining a core with an appropriate size). 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). Applying this restriction reduces the number of named entities to an average of 12 per topic 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 [18] 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 algorithms to determine the path along which the greatest amount of activation will travel between the two nodes, which we 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 primary 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). As an example, the core obtained from the “Elian Gonzalez Custody Battle” network (See Figure 5) maintained the primary relationships between the important entities within the network, but discarded information such as the dates of many trials, the quotes of less important figures relating to the case, and information about entities which did not directly relate to the case itself.

Running the algorithm on each of the topic 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, since the resulting core tends to contain

the most salient nodes and relations in the network human evaluators can 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.

6.2.2. *Evaluating the Core*

ASKNet uses the GraphViz [19] 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. Examples of the graphical representations of the network cores used for evaluation are shown in Figure 5 and Figure 7. Magnified views of the representations are also given in Figure 6 and Figure 8. The representations are similar to those 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). Each evaluator was 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.^a

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 as shown in Figure 9), or a *label error* (an incorrect set of labels, indicating that the node did not correspond to a single real world entity as shown in Figure 10). The evaluators were then asked to evaluate each primary path. If there was an error at any point in the path, the entire path was said to have a *path error* (as shown in the Figure 11) and deemed to be incorrect. In particular, note that in the bottom example of Figure 11, the error caused several paths (i.e., “Melissa Virus” - “Microsoft Word”, “Melissa Virus” - “Microsoft Word documents” and “Microsoft Word” - “Microsoft Word documents”) to be considered

^aAll of the evaluation materials provided to the evaluators can be downloaded at: <http://www.brianharrington.net/asknet>.

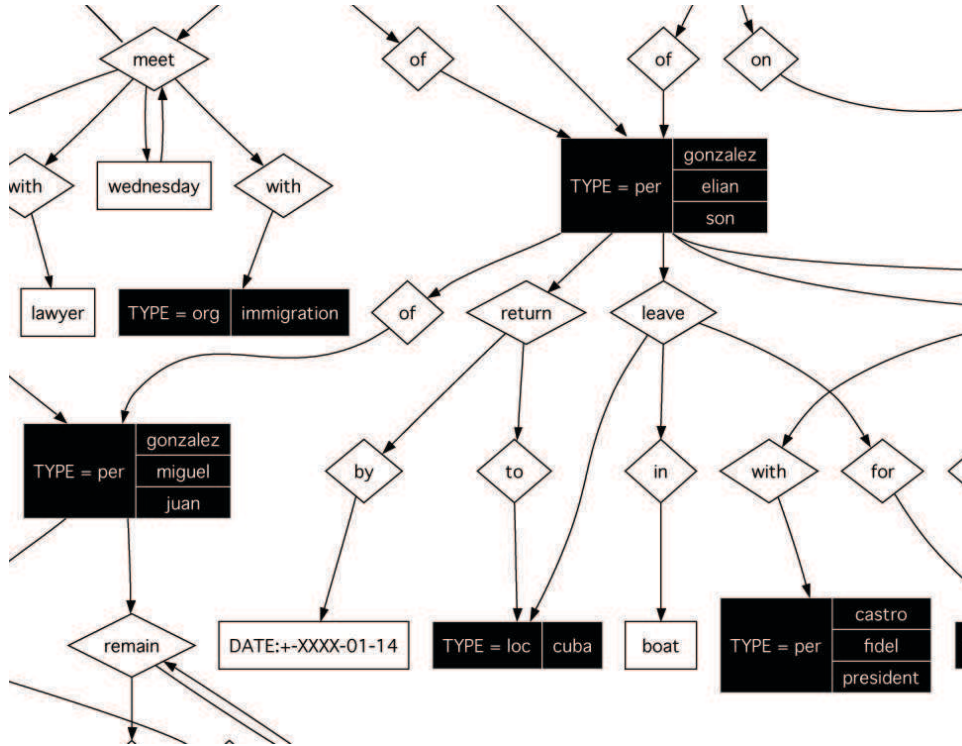


Fig. 6. Expanded section of Figure 5

| 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 5. Evaluation Results

from a single source, but rather are scattered across each of the steps. The *NE Type* errors were made by the 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

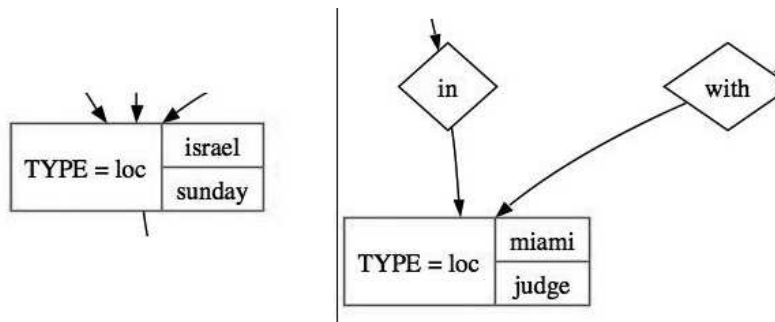


Fig. 10. Examples of *label errors*. Left: After processing a sentence containing the phrase “...arriving in Israel Sunday to conduct...” the phrase “Israel Sunday” is mistakenly identified as a single entity. Right: The location “Miami” and the person “Miami Judge” collapsed into a single node.

| 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 6. Errors by Type

7. Conclusion

The ASKNet system is able to quickly generate large scale integrated semantic networks from multiple natural language resources. Natural language text represents a potential gold mine of semantic knowledge, and is available in large quantities. ASKNet is able to mine some of that knowledge using current NLP technologies, and then to use spreading activation algorithms to integrate that knowledge into a single cohesive resource. In this paper 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.

ASKNet uses semantic networks to represent its knowledge since they are general semantic resources with potential uses in a wide variety of applications. The enormous manual effort given to projects such as Cyc demonstrates the need for such networks. The innovation of the ASKNet system is the design of a method for efficiently generating large scale, expressive and well integrated semantic networks. ASKNet was able to create a network twice as large as Cyc in less than 3 days.

Evaluating semantic networks is a difficult task. In order to evaluate ASKNet, we developed a novel metric based on human evaluators measuring the precision

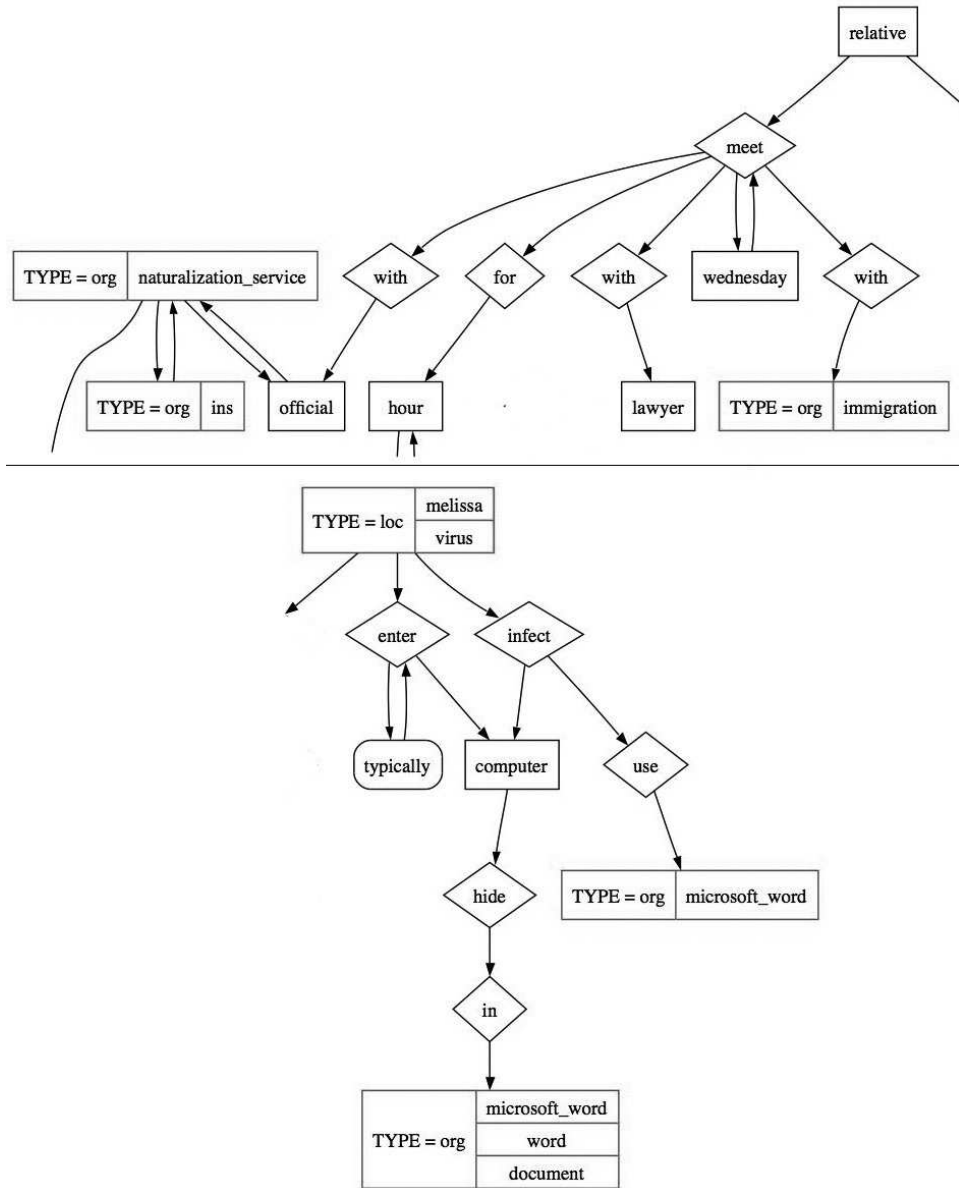


Fig. 11. Examples of *path errors*. Top: Three independent meetings referenced in the same sentence, all involving relatives of Elian Gonzalez, are identified as a single meeting event. Bottom: The network indicates that the computer, rather than the virus, hides in Microsoft Word documents.

of the network core. Using this metric we obtained results of almost 80% for five ASKNet networks created from randomly chosen DUC articles.

There are many potential uses of large-scale semantic networks. In this paper we focus on the construction and evaluation of such networks, rather than their application. However, one particularly compelling application for future work is novel fact discovery: using the spreading activation algorithm to provide a semantic distance measure between entities, allowing the discovery of novel (and perhaps surprising) connections between entities in the network. In particular, finding entities which have connections within the network, but do not co-occur within documents, could lead to the discovery of facts and relationships that would otherwise be difficult or impossible to find. This research could have an impact on many fields, particularly in the biomedical domain. The ability to discover relationships between genes and proteins, or chemicals and diseases that are not currently known to be connected could have profound and far-reaching consequences in the fields of biomedicine, disease treatment and genetics.

Acknowledgements

We would like to thank the evaluators for taking the time to provide us with an evaluation of the networks, as well as the initial test subjects whose feedback was instrumental in setting up the evaluation experiments. We would also like to thank the anonymous reviewers of the International Conference on Semantic Computing (ICSC08) for their helpful feedback on the initial conference paper which was the basis for this work. This work was funded in part by scholarships from the Clarendon Fund, and the Canadian Centennial Scholarship fund.

References

- [1] H Liu and P Singh. ConceptNet a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22:211 – 226, Oct 2004.
- [2] D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33 – 38, 1995.
- [3] Cynthia 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.
- [4] Jon Curtis, G. Matthews, and D. Baxter. On the effective use of Cyc in a question answering system. In *Papers from the IJCAI Workshop on Knowledge and Reasoning for Answering Questions*, Edinburgh, Scotland, 2005.
- [5] Tom Stocky, Alexander Faaborg, and Henry Lieberman. A commonsense approach to predictive text entry. In *Proceedings of Conference on Human Factors in Computing Systems*, Vienna, Austria, April 2004.
- [6] William B. Dolan, L. Vanderwende, , and S. Richardson. Automatically deriving structured knowledge base from on-line dictionaries. In *Proceedings of the Pacific Association for Computational Linguistics*, Vancouver, British Columbia, April 1993.
- [7] Stephen D. Richardson, William B. Dolan, , and Lucy Vanderwende. MindNet: Acquiring and structuring semantic information from text. In *Proceedings of COLING '98*, 1998.

- [8] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics (COLING/ACL-06)*, pages 113–120, Sydney, Australia, 2006.
- [9] L. Schubert and M. Tong. Extracting and evaluating general world knowledge from the brown corpus. In *Proceedings of the HLT/NAACL 2003 Workshop on Text Mining*, 2003.
- [10] S. Clark and J. R. Curran. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552, 2007.
- [11] Mark Steedman. *The Syntactic Process*. The MIT Press, Cambridge, MA., 2000.
- [12] J. Hockenmaier. *Data and Models for Statistical Parsing with Combinatory Categorical Grammar*. PhD thesis, University of Edinburgh, 2003.
- [13] 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.
- [14] Johan Bos, Stephen Clark, Mark Steedman, James R. Curran, and Julia 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.
- [15] 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.
- [16] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.
- [17] D.E. Meyer and R.W. Schvaneveldt. Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, 90(2):227–234, 1971.
- [18] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerical Mathematics*, 1:269 – 271, 1959.
- [19] 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.
- [20] J. Carletta. Assessing agreement on classification tasks: the Kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.