Computing Science Group

# Conceptual Knowledge Acquisition Using Automatically Generated Large-Scale Semantic Networks

**Pia-Ramona Wojtinnek**
**Brian Harrington**
**Sebastian Rudolph, Institut AIFB Karlsruhe, DE**
**Stephen Pulman**

CS-RR-10-04

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford, OX1 3QD

**Abstract.** We present a method for automatically creating large-scale semantic networks from natural language text, based on deep semantic analysis. We provide a robust and scalable implementation, and sketch various ways in which the representation may be deployed for conceptual knowledge acquisition. A translation to RDF establishes interoperability with a wide range of standardised tools, and bridges the gap to the field of semantic technologies.

## 1 Introduction

Graph-based models for representing conceptualisations have a long-standing history, ranging from expressive logical frameworks (as laid out in Peirce's work and further developed into conceptual graphs [1]) to widely applied graph-based Semantic Web formalisms like the Resource Description Framework (RDF) [2]. Graph-based representations of knowledge have been shown to provide both intuitive and formally rigorous access to the represented information.

In the variety of graph-based KR formalisms, two directions can be identified, reflecting the separation of AI approaches into associative/statistic and symbolic. These differing perspectives also show up when addressing the task of processing natural language for explicating the implicit conceptual information contained in a textual resource. Depending on the output of such an elicitation step and on the way the result is used, we can roughly distinguish between primarily symbolic approaches (such as categorial grammars or discourse representation theory) and statistical ones (such as vector space models, or hidden Markov models). This distinction carries over to the case when natural language is transformed into graph-based representations.

Research into symbolic approaches has resulted in systems which are able to process natural language text to create Conceptual Graphs and use them for scenario recognition [3–5]. However, complete correctness of knowledge acquired from text cannot be guaranteed, while at the same time, logical representations are inherently extremely sensitive to flawy data.

Statistically oriented approaches, using representations such as association networks, and co-occurrence graphs are more tolerant to the small but inevitable errors in the acquisition process, and can naturally deal with noisy data. However, there is a great deal of valuable information unveiled by a syntactical and discourse-theoretic analysis even if the resulting structure is meant to be deployed in a statistic way. In general, recent developments in statistic text analysis witness a stronger appreciation of symbolic NLP preprocessing.

Hence, we advocate a graph-based conceptual model which provides a semantic middleground: while it exhibits structural dependencies way beyond mere co-occurrence, it still features a fault tolerant way of representing the conceptual semantics of the original textual resource rather than providing a crisp logical description.

We further develop the ASKnet system [6] for conceptual knowledge acquisition and representation. AskNet provides a framework for acquisition of graph-based semantic networks from English text and is described in Section 2. In Section 3, we describe the additional steps that allow us to apply the framework to the problem of representing the conceptual backbone of a given text corpus in an aggregated yet structurally informative way. Furthermore, in Section 4, we provide a translation of the described graph model into RDF and sketch the plethora of benefits that arise from the interoperability achieved by the alignment with this wide-spread, standardised, graph-based Semantic Web KR formalism.

## 2 AskNet

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.

The C&C toolset [7] is used to parse and analyse text before it is used to create ASKNet networks. This allows ASKNet to create networks using entities and relationships taken directly from naturally occurring text, and allows for an almost limitless variety of relations. The C&C toolset contains an efficient and robust dependency parser and named entity recogniser, as well as a semantic analysis tool called Boxer [8], which converts the parsed text into a series of first-order logic predicates in a discourse representation structure. Figure 1 gives an example of Boxer output, and the corresponding ASKNet network fragment.

ASKNet is capable of efficiently generating integrated semantic networks on a scale never seen before. In [6] the speed and robustness of ASKNet was demonstrated by creating networks consisting of over 1.5 million nodes and 3.5 million edges — more than twice as large as any manually created semantic network — in less than 3 days.

Large scale networks such as ConceptNet [9] and WordNet [10] have been shown to be useful for question answering [11] and predictive text entry [12]. However, ASKNet is unique in that it can construct networks directly from natural language text, and extracts its relations directly from the text, rather than relying on a set of pre-defined relations. This results in networks which can be created and adapted to new domains very rapidly, while at the same time allowing for an extremely wide variety of relations.

### 2.1 The Network

The semantic networks created by ASKNet consist of object nodes and attribute nodes linked by directed labelled relations. Object nodes (rectangle) correspond roughly to discourse referents (e.g., nouns, pronouns), attribute nodes (oval) correspond to modifiers (e.g., adjectives and adverbs) and relation nodes (diamond) correspond to events and binary relations.

The nested structure of the network allows for concepts and relations to be combined to form more complex concepts, allowing for arbitrary complexity. An example network is given in Figure 1.

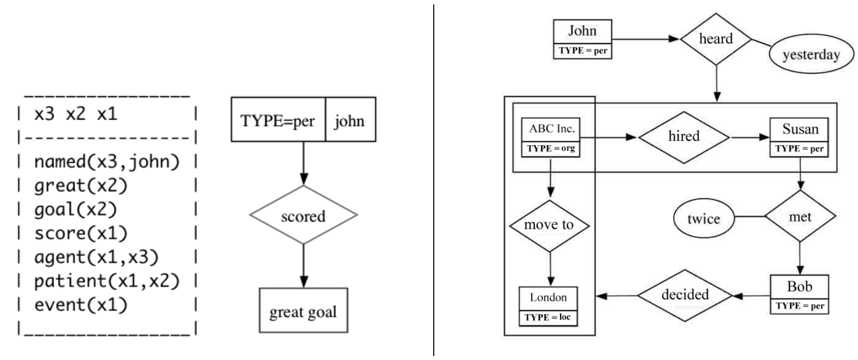



**Fig. 1.** Left: Boxer output and corresponding network fragment for the sentence "John scored a great goal." Right: A simplified Semantic Network created from the sentences "Yesterday John heard that ABC Inc. hired Susan. Bob decided that ABC Inc. will move to London. Susan met Bob twice."

### 2.2 Spreading Activation

ASKNet builds on the concept of relationships in the human brain, which are semantically linked [13] so that thinking about (or firing) one concept primes other related concepts making them more likely to fire in the near future. Each node within the network can receive activation. If this activation reaches a set threshold the node will fire, sending its activation to its neighbours, with the proportions dictated by the strength of the connecting relationships. This can in turn cause other nodes to fire, spreading activation throughout the network.

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. Nodes which send similar amounts of activation to similar sets of concepts are determined to be semantically related. The details of the spreading activation algorithm can be found in [6]

### 2.3 Information Integration

One of the most important aspects of ASKNet is the way in which it integrates information from different sources in order to create a single cohesive network. This process, called the *update algorithm*, uses ASKNet's spreading activation algorithm, combined with lexical information stored in the nodes to decide which nodes refer to the same real world object, and therefore should be mapped together.

In the example shown in Figure 2, ASKNet attempts to integrate the network fragment relating to George Bush's 2000 U.S. election victory into an existing network. ASKNet first uses string and named entity similarity to produce an initial mapscore for the mappings between nodes. In our example, the mapscore (*bu*,*georgebush*) and (*bu*,*johnbush*) would likely be approximately equal, as in both cases they are of the same named entity type, and all labels in both target nodes contain the label of the

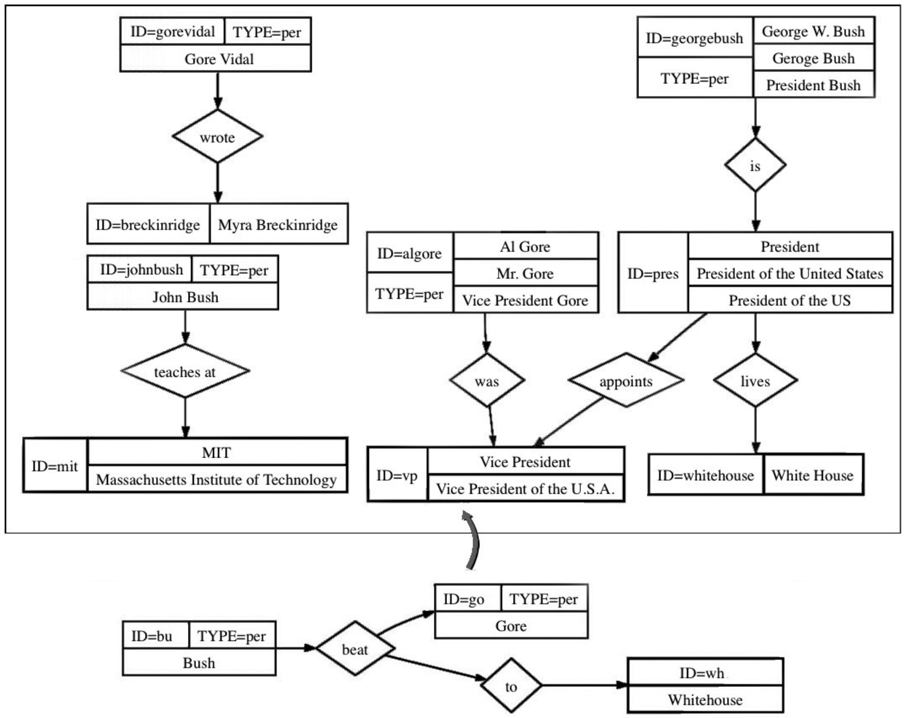



**Fig. 2.** An example update network created from the sentence "Bush beat Gore to the Whitehouse" being added to a network containing information about United States politics, along with some extraneous information regarding famous authors and university lecturers.

source node. In order to improve these scores, the update algorithm is run. We will briefly walk through an example of steps taken by the update algorithm to refine the relative scores for the *bu* node.

The source node (*bu*) is provided with activation and fired. Activation spreads through the network fragment, with various amounts ending in each of the test nodes (*go* and *wh*). The activation in each of those nodes is then transferred to the main network. In this case, the activation from *wh* will be sent to *whitehouse*, and the activation from *go* will be split between *gorevidal* and *algore* depending on their current mapscore. For example, if the current mapscore for (*go*,*gorevidal*) is twice that of (*go*,*algore*), then *algore* will recieve $^1/_3$ of the activation and *gorevidal* will receive $^2/_3$.

Once the activation is placed into the main network and allowed to fire, activation will be spread to connected nodes. All mapscores of the source node are then updated. In our example, the score (*bu*,*georgebush*) will likely be increased, as *georgebush* will have received activation, whereas (*bu*,*johnbush*) will decrease as *johnbush* received no activation.

The update algorithm becomes a reinforcing loop, gradually improving scores for mappings that are likely to be connected, and decreasing scores for unrelated pairs. Eventually, pairs with a score above a set threshold are mapped together, thus integrating the new fragment into the existing network. In our example, this would map *bu* to *georgebush* and *go* to *algore*. Further details of the update algorithm are given in [6].

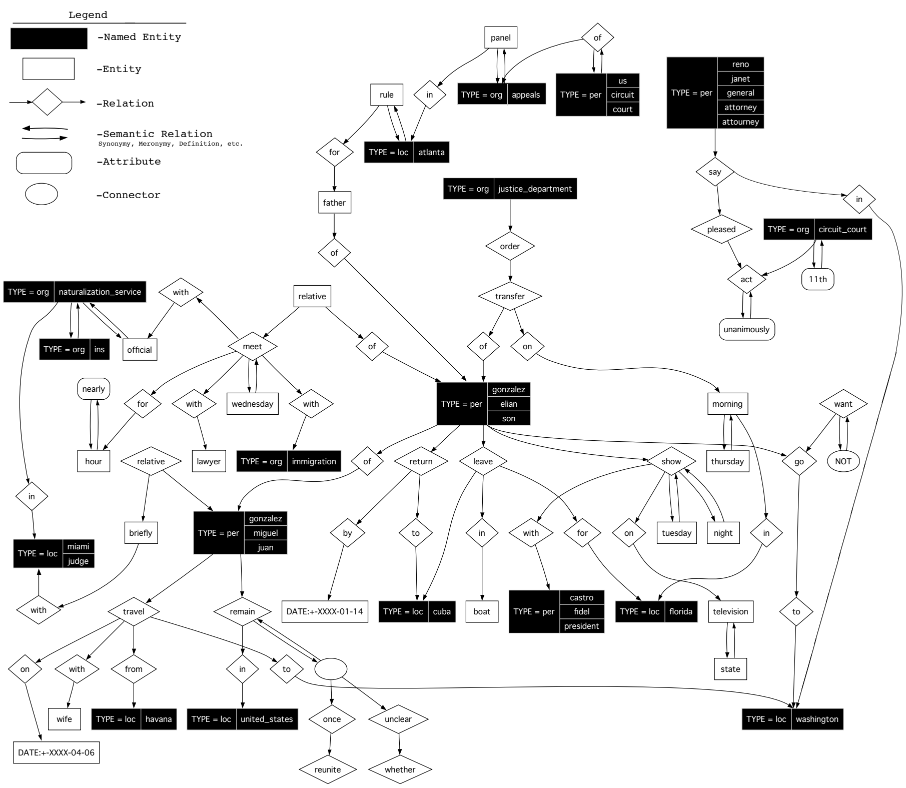



**Fig. 3.** Graphical representation for topic: "Elian Gonzalez Custody Battle".

## 2.4 Evaluation

**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 [6]; a vast improvement over manually created networks for which years or even decades are required to achieve networks of less than half this size [14]. 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 the size of the network ceases to have an effect on the algorithm. Therefore the average time to add a new node to the network is asymptotic and allows the integration process to scale easily to extremely large networks.

**Network Precision:** In [6] networks were built using documents from the 2006 Document Understanding Conference (DUC), and the "core" of each network was extracted for evaluation. Examples of the graphical representations of the network cores used for evaluation are shown in Figure 3.

These network cores were given to human evaluators, who were asked to label each path, node and named entity tag within the network as correct or incorrect. An overall average precision of 79.1% was assigned by the judges [6], which is highly promising for such a difficult task.

## 3 Developments for Conceptual Knowledge Acquisition

We propose to develop the AskNet approach further for conceptual knowledge acquisition. This involves the development of both the building processes as well as the processing of the resulting network. The core idea is to integrate the information on concepts, instead of Named Entities, in order to construct a network representing concepts and relations between them.

Figure 4 shows a subgraph of a sample network, which was built from a few paragraphs taken from Graduate Studies Handbooks. Multiple occurrences of the same relations have been pictured as overlapping. As this was a small and domain-specific text collection, it was not necessary to distinguish between different senses of a term and thus the integration algorithm simply mapped object nodes with the same label together. We will discuss word sense disambiguation in the next section.

The resulting structure of the network is as follows. From one object node to the next (i.e., without interfering object nodes) are relations which syntactically occurred in the text, such as *student complete dissertation* and *progress of student*. Due to the use of Boxer, these relations are normalised, e.g., the latter relation may have been expressed as *student's progress* in the text. As can be seen, the frequencies of these relations provide ground for weighting. The details of the weighting depend on the further application. For example, the *complete* relations going from *student* to *credit*, *module*, *dissertation* and *work* can either be kept separately or be joined together, resulting in one subject link from *student* to *complete* and four object links from *complete* to the others with according weighting.

Looking at the whole semantic network, it provides a dense and interconnective representation of the concepts and their relations in a cross-sentence and cross-document way. For example, out of the 120 and 180 occurrences of *dissertation* and *module* in the text, respectively, the two co-occurred in a sentence only 8 times. However, the network shows that there is a strong connection between the two concepts, as they are both strongly and directly connected to the concept student. In particular, the connecting relations are explicitly specified and can hence be directly identified.

In the next section we will sketch ways in which the building and integration process as well as the resulting network can be used for conceptual knowledge acquisition.

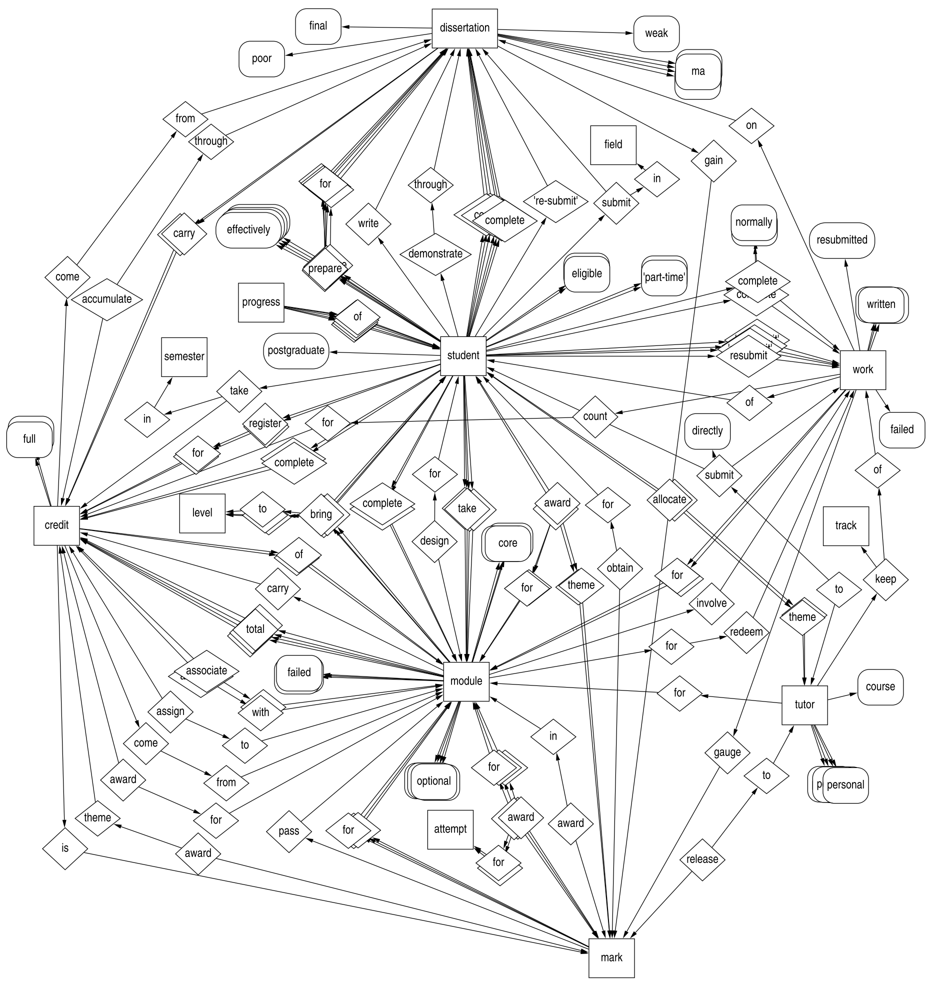



**Fig. 4.** Subgraph displaying selected concepts and relations from sample network.

### 3.1 Word Sense Induction and Disambiguation

One of the core challenges in conceptual knowledge acquisition from text is the gap between terms and concepts. Terms can be polysemous, (e.g., *bank* can refer to a river bank as well as to a financial institution) and on a more finely grained level, terms often have sense variants, for example *bank* referring the bank building. Therefore, in order to acquire knowledge about concepts from text, it is crucial to both identify different senses of a term and disambiguate instances of the term in text.

The currently best performing Word Sense Disambiguation (WSD) techniques are supervised in two respects [15]. Firstly, they are given a sense inventory such as WordNet [16] which has been built beforehand and provides a list of possible senses of each term. Secondly, they are trained on a corpus such as SemCor [17], which has been hand-tagged with the senses of the inventory. This leads to certain limitations. The large amount of hand-tagged data needed means that the improvement of the performance of general WSD relies to a great extent on the extension of these resources. In addition, the adaptation of supervised techniques to specific domains may require additional annotated corpora. Similarly, domain-specific sense inventories are not readily available.

Unsupervised Word Sense Induction and Discrimination (WSID) attempts to tackle these issues [15]. The goal is to induce word senses directly from the corpus in an unsupervised manner. Most work in WSI is based on the vector space model of meaning. For a target word, the contexts in which it occurs are represented as vectors. These are then clustered using vector similarity measures, giving different senses of the word. Once established, a new instance of the term can be classified by measuring its similarity to the clusters.

Different graph-based models have been used for WSD and WSID. Many approaches use a machine readable dictionary such as WordNet to build a graph representation [18–22]. WordNet provides senses of terms in the form of synsets, semantic relations between synsets (e.g., hyponymy and meronymy) and a gloss for each sense. This network-like structure can be translated into graphs in various ways, depending how much and which information is used. Recently, the network structure provided by Wikipedia links [23, 24] has also been exploited for WSD. Rather than relying on a pre-existing network resource, approaches such as those described in [25–27] build co-occurrence networks directly from text.

As we build networks directly from text, our approach is most related to those which use co-occurrence networks. However, our representation is semantically richer. In addition, the update algorithm provides us with a new method of network building and use. In the next two sections we sketch different ways in which our approach can be used for WSI and WSID.

**Word Sense Induction** Two strategies could be followed to use the semantic network for unsupervised Word Sense Induction.

The first strategy uses the update algorithm described in Section 2 to detect different senses of terms. Previously, the update algorithm aimed to have exactly one node per named entity in the network, merging all occurrences of the entity into this node. For WSI, the aim is to have one node per word sense, merging into that node all occurrences of the word used in that particular sense.

The update algorithm is used to decide which mapping is most suitable or to create a new node if none of them reaches a certain threshold. The spreading activation setting allows for finely-grained parameter tuning, which can be exploited. For example, different types of nodes and links can be set to contribute more or less in the process. The update algorithm can also take into account relation labels, which can potentially improve the network's disambiguation ability.

A network constructed in this manner can then be used for word sense disambiguation. Following a similar approach, the update algorithm can assign words in a sentence to a particular word sense, and return the mapping that would have been created during the building phase.

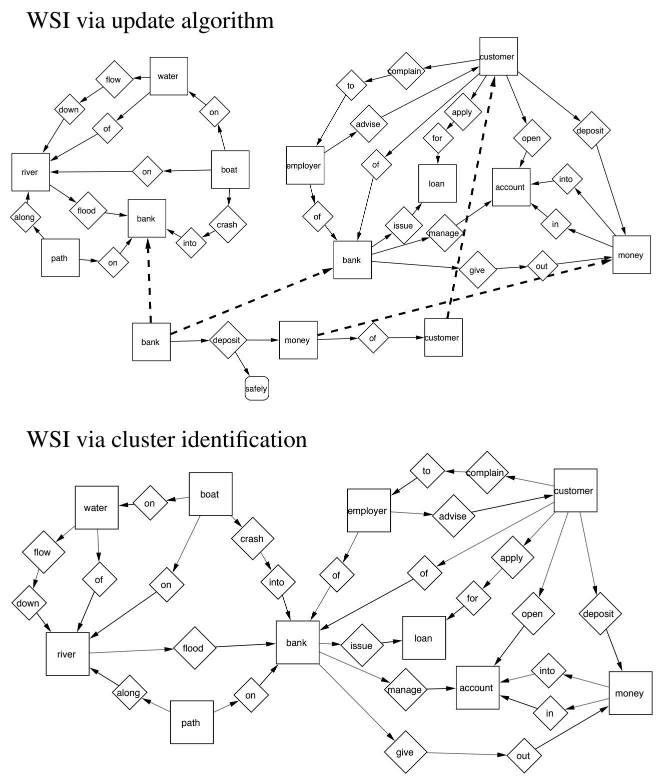



**Fig. 5.** Illustrating examples for the two WSI strategies.

The second strategy does not disambiguate during the construction of the network but simply merges all occurrences of a word into one node while building the network from a corpus. The rationale is that we expect the network around a polysemous word to show subclusters, corresponding to different contexts in which the word is used. These subclusters of context in turn represent different senses. Once the clusters are identified, new instances can be classified using a similarity measure between the context of the instance and the clusters established.

This approach is in line with [25], who showed that co-occurrence graphs build from a collection of paragraphs in which a target word appears exhibit small world strucutre, i.e., they contain the expected highly dense subgraphs corresponding to different senses.

**Supervised WSD and Domain Adaptation** ASKNet networks can also be used for Supervised Word Sense Disambiguation. Using a tagged corpus, a network is built integrating nodes with the same sense tag. New instances can then be disambiguated using the update algorithm as described in the previous section.

In particular, the semantic network built from a general, sense-tagged corpus such as SemCor could provide a ground for semi-supervised domain adaptation. This scenario of domain adaptation supplements a supervised, generic WSD with untagged examples from the specific domain [28]. Good results on that task would show that supervised generic systems can be tweaked at little cost to be more suitable for domain specific WSD. A network built from a generic corpus can easily be extended by feeding domain specific text into the integration process. We would expect some senses of a term to be further extended by adding domain specific documents, and other senses to appear as new nodes, leading to a domain-tuned disambiguation network.

### 3.2 Network as Context

The second group of applications exploits the network as a representation of meaning. In NLP, one of the most common representations of the meaning of a term is in the form of a vector based on the context in which the term appears. Similarity of terms can then be computed using vector similarity measures [29]. The vector space model has been used in a variety of tasks relevant to conceptual knowledge acquisition, in particular in the field of ontology learning [30]. Examples are term similarity, hierarchical and non-hierarchical clustering.

Vectors can be built in different ways, depending on what is considered the context of a term. In its simplest form, the vector is built from the words which co-occur with the target word in a document, paragraph or window. Co-occurrence based vectors have been widely used, but with fast and robust syntactic parsers becoming available, there is a great deal of current work in exploiting more refined vector representations, extracting words which stand in a syntactic relationship to the target word [31–33, 29]. Most approaches take several syntactic relationships into account, collecting triples such as *banana AdjMod tasty* and *banana dirObj eat*, but the restriction to specific relations such as attributes is also possible [34].

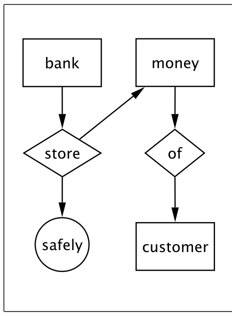
We propose to construct the vector space representation for a target word using the semantic network as context. Not only will this allow us to use the syntactic and semantic dependencies connected to our node, but also to exploit the information contained in the wider network. For example, in Fig. 4 the network surrounding *module* contains the syntactic relations *student take module*, *module carry credit* and *mark for module*. This would provide similar information to that used for the syntactic vector space models described above. However, the network structure provides a far richer context, including relation strengths and complex, multi relation chains. Thus, not only do we know that *module* is syntactically connected to *credit* and *student*, but *credit* and *student* are connected through *complete*, and that both *credit* and *student* are connected to *dissertation*.

A vector representation of the context of a target word in the network can be constructed using spreading activation. In [6] we showed that using the activation received by a node when its prime is fired outperforms purpose-built vector space models on predicting priming effects in human subjects, and performs as well as top scoring systems in judging the semantic relatedness of terms. The spreading activation relatedness measure reflects the interconnections between objects over the network. In our example, the activation *student* receives directly from *module* would be reinforced by activation from *module* via *credit* and *work* and *dissertation* would indirectly receive activation through *credit* and *student*. Therefore, by activating the target node and constructing the vector from the nodes that have received activation with their activation scores as values we can leverage the rich network structure to create a more robust vector-based representation.

This approach can be tested in several task which have used vector representation, such as finding synonyms and conceptual clustering.

## 4 RDF Serialisation

We have implemented a serialisation of our graph model into the Semantic Web Resource Description Framework.[1] As RDF itself is a graph-based datamodel, a translation is rather straightforward. Node types are explicitly indicated via `rdf:type` and node labels are associated as untyped literals via `rdf:label`.[2] Table 1 shows a small graph and its translation into RDF.[3]




```
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix x:        <http://example.org/> .
@prefix link:     <http://example.org/link/> .
@prefix nodeType: <http://example.org/nodeType> .
x:id1   rdf:type      nodeType:Object ;    rdf:label "bank" .
x:id2   rdf:type      nodeType:Event ;     rdf:label "store" .
x:id3   rdf:type      nodeType:Attribute ; rdf:label "safely" .
x:id4   rdf:type      nodeType:Object ;    rdf:label "money" .
x:id5   rdf:type      nodeType:Prep ;      rdf:label "of" .
x:id6   rdf:type      nodeType:Object ;    rdf:label "customer" .
x:id1   link:subLink  x:id2 .
x:id2   link:attLink  x:id3 .
x:id2   link:objLink  x:id4 .
x:id4   link:subLink  x:id5 .
x:id5   link:objLink  x:id6 .
```

**Table 1.** Graph and its RDF serialisation.

By bridging the gap to the Semantic Web world, a variety of RDF tools can be used. RDF databases (so-called *triple stores*, such as Jena (http://jena.sourceforge.net/)) can be used for storage and querying of large-scale networks. In particular, with SPARQL [36], we are endowed with a standardised querying language allowing for retrieval of graph patterns and – by virtue of query output via `CONSTRUCT` – on-the-fly creation of new networks. As an example, the following SPARQL query would produce a new graph containing all object nodes that are linked to nodes with label ”student” over a preposition node:

```
CONSTRUCT * WHERE { ?x ?p ?y . ?p rdf:type nodeType:prep .
                    ?x rdf:type nodeType:Object ; rdf:label ?xn .
                    ?y rdf:type nodeType:Object ; rdf:label ?yn .
                    FILTER (?xn="student" || ?yn="student") }
```

[1] We assume the reader to be familiar with the basics of the RDF language (cf. [35] for an introduction into RDF and SPARQL).

[2] In order to incorporate edge weights in the representation, some more considerations have to be taken: encoded in the natural way, weighted labeled edges correspond to a 4-ary relationship ⟨`source`, `label`, `target`, `weight`⟩. However, this issue can be solved by a well-established best practice for such situations in RDF called *reification*.

[3] Note that for maximal interoperability, the actual output generated by our system follows the RDF/XML serialisation. However, for the sake of brevity and understandability as well as a better alignment with the SPARQL notation, we use Turtle syntax in our examples.

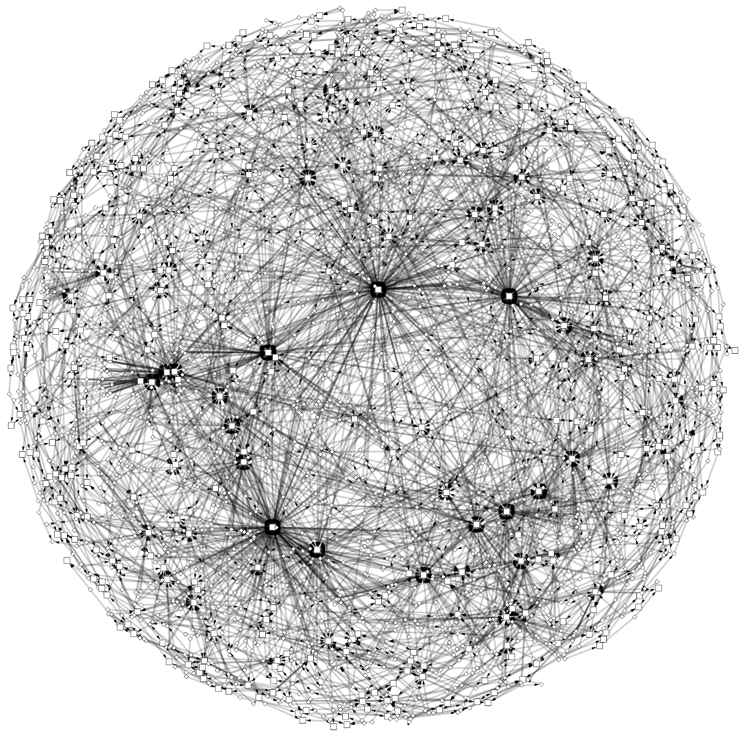


**Fig. 6.** Complete network representation of Fig. 4 showing hubs.

This query would create the union of graph fragments corresponding to phrases like "progress of student" or "student in department".

Moreover RDF-compatible graph-drawing tools with versatile layout libraries greatly facilitate to visualise and explore the networks. They can make a variety of information immediately visible, such as hubs or clusters. For instance, Fig. 6 was created with the RDFScape plugin for the graph-drawing tool Cytoscape (http://www.cytoscape.org/), which on top of its visualisation functionality also provides intuitive methods for filtering, navigating and manipulating large graphs.

Beyond the advantage of a lot of tools becoming available, providing an RDF interface also enables interoperability on the level of resources. In the course of the Linked Data initiative (http://linkeddata.org/), more and more data from various domains is made publicly available, with RDF taking the role of a *lingua franca* of data exchange and syndication. External resources – no matter whether lexical (e.g., Wordnet [37]), encyclopedic (as in DBpedia (http://wiki.dbpedia.org)) or highly structured ontological ones (such as OpenCyc (http://sw.opencyc.org/) or DOLCE (http://www.loa-cnr.it/DOLCE.html)) – can be easily accessed and integrated with graph models that have been created by our approach, enabling intense usage of background knowledge of any kind and countering potential problems with graph sparseness.

## 5 Conclusion

We have presented an approach for building large scale semantic networks automatically from text, employing deep semantic processing. Our graph model provides a well-balanced middle ground between purely symbolic and numerical approaches to graph-based knowledge representation. We have identified several ways in which our semantic network models can be used for conceptual knowledge acquisition. Our implementation of the building algorithm is highly competitive in terms of coverage and performance. Providing an encoding into the RDF standard allows us to employ semantic technologies for conveniently managing large networks.

Future work includes a rigorous evaluation in order to investigate the added value of our approach compared to other graph representations generated from text such as co-occurrence graphs, resources such as ConceptNet and WordNet as well as task specific non graph-based methods. We aim to evaluate our framework in several scenarios, two of which have been described above – Word Sense Induction and Disambiguation as well as further tasks typically addressed by a vector space representation of meaning.

## References


1. Sowa, J., ed.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading, MA (1984)
2. Manola, F., Milner, E.: RDF Primer. W3C Recommendation (10 Februar 2004) Available at http://www.w3.org/TR/REC-rdf-syntax/.
3. Boytcheva, S., Dobrev, P., Angelova, G.: Cgextract: towards extraction of conceptual graphs from controlled english. In Mineau, G., ed.: Contributions to ICCS-2001. Volume 41., CEUR Workshop Proceedings (2001) 89–116
4. Boytcheva, S., Yankova, M., Strupchanska, A.: Semantically driven approach for scenario recognition in the ie system fret. In Nicolov, N., Bontcheva, K., Angelova, G., Mitkov, R., eds.: RANLP. Volume 260 of CILT., Benjamins (2003) 257–266
5. Yankova, M., Boytcheva, S.: Focusing on scenario recognition in infomation extraction. In: EACL. (2003) 41–48
6. Anonymous: Anonymous citation. Removed
7. Clark, S., Curran, J.R.: Wide-coverage efficient statistical parsing with CCG and log-linear models. Computational Linguistics **33**(4) (2007) 493–552
8. Bos, J., Clark, S., Steedman, M., Curran, J.R., Hockenmaier, J.: Wide-coverage semantic representations from a CCG parser. In: COLING '04, Geneva, Switzerland (2004) 1240–1246
9. Liu, H., Singh, P.: Conceptnet: A practical commonsense reasoning tool-kit. BT Technology Journal **22** (Oct 2004) 211 – 226
10. Fellbaum, C., ed.: WordNet : An Electronic Lexical Database. MIT Press, Cambridge, Mass, USA (1998)
11. Curtis, J., Matthews, G., Baxter, D.: On the effective use of Cyc in a question answering system. In: IJCAI Workshop on Knowledge and Reasoning for Answering Questions. (2005)
12. Stocky, T., Faaborg, A., Lieberman, H.: A commonsense approach to predictive text entry. In: CHI'04, Vienna, Austria (April 2004)
13. Meyer, D., Schvaneveldt, R.: Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. Journal of Experimental Psychology **90(2)** (1971) 227–234

14. Matuszek, C., Cabral, J., Witbrock, M., DeOliveira, J.: An introduction to the syntax and content of Cyc. In: AAAI SS on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering. (2006)
15. Agirre, E., Soroa, A.: Semeval-2007 task 02: Evaluating word sense induction and discrimination systems. In: SemEval '07, ACL (June 2007) 7–12
16. Fellbaum, C.: WordNet: An Electronic Lexical Database. The MIT Press (1998)
17. Miller, G.A., Leacock, C., Tengi, R., Bunker, R.T.: A semantic concordance. In: HLT '93, ACL (1993) 303–308
18. Veronis, J., Ide, N.M.: Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In: COLING '90, ACL (1990) 389–394
19. Sussna, M.: Word sense disambiguation for free-text indexing using a massive semantic network. In: CIKM '93, ACM (1993) 67–74
20. Mihalcea, R., Tarau, P., Figa, E.: Pagerank on semantic networks, with application to word sense disambiguation. In: COLING '04, ACL (2004) 1126
21. Navigli, R., Velardi, P.: Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. IEEE Trans. Pattern Anal. Mach. Intell. **27**(7) (2005) 1075–1086
22. Tsatsaronis, G., Vazirgiannis, M., Androutsopoulos, I.: Word sense disambiguation with spreading activation networks generated from thesauri. In: IJCAI'07, Morgan Kaufmann (2007) 1725–1730
23. Mihalcea, R.: Using wikipedia for automatic word sense disambiguation. In: HLT-NAACL. (2007) 196–203
24. Fogarolli, A.: Word sense disambiguation based on wikipedia link structure. In: IEEE Int. Conf. on Semantic Computing. (2009)
25. Véronis, J.: Hyperlex: lexical cartography for information retrieval. Computer Speech and Language **18** (2004)
26. Agirre, E., Martínez, D., de Lacalle, O.L., Soroa, A.: Two graph-based algorithms for state-of-the-art wsd. In: EMNLP '06, ACL (2006) 585–593
27. Klapaftis, I.P., Manandhar, S.: Uoy: a hypergraph model for word sense induction & disambiguation. In: SemEval '07, ACL (2007) 414–417
28. Agirre, E., de Lacalle, O.L., Fellbaum, C., Marchetti, A., Toral, A., Vossen, P.: Semeval-2010 task 17: all-words word sense disambiguation on a specific domain. In: DEW '09: Proc. WS on Semantic Evaluations: Recent Achievements and Future Directions, ACL (2009) 123–128
29. Padó, S., Lapata, M.: Dependency-based construction of semantic space models. Comput. Linguist. **33**(2) (2007) 161–199
30. Biemann, C.: Ontology learning from text: A survey of methods. LDV Forum **20**(2) (2005) 75–93
31. Grefenstette, G.: Explorations in automatic thesaurus discovery. Kluwer (1994)
32. Lin, D.: Automatic retrieval and clustering of similar words. In: COLING'98, ACL (1998) 768–774
33. Curran, J.R., Moens, M.: Improvements in automatic thesaurus extraction. In: Proc. ACL-02 WS on Unsupervised lexical acquisition, ACL (2002) 59–66
34. Almuhareb, A., Poesio, M.: Attribute-based and value-based clustering: An evaluation. In: EMNLP '04, ACL (2004) 158–165
35. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
36. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF. W3C Recommendation (15 Jan. 2008) Available at http://www.w3.org/TR/rdf-sparql-query/.
37. van Assem, M., Gangemi, A., Schreiber, G.: RDF/OWL Representation of WordNet. W3C Working Draft (19 June 2006) Available at http://www.w3.org/TR/wordnet-rdf/.