Construction of Conceptual Graph representation of texts

Svetlana Hensman
Department of Computer Science
University College Dublin
Belfield, Dublin 4
svetlana.damianova@ucd.ie

Abstract

This paper describes a system for constructing conceptual graph representation of text by using a combination of existing linguistic resources (VerbNet and WordNet). We use a two-step approach, by firstly identifying the semantic roles in a sentence, and then using these roles, together with semi-automatically compiled domain-specific knowledge to construct the conceptual graph representation.

1 Introduction

The problem of automatic acquisition of knowledge is an interesting and challenging one and has been tackled by linguists for some time.

This paper describes a system for automatic conceptual graph acquisition using a combination of linguistic resources, such as VerbNet and WordNet, together with semi-automatically compiled domain-specific knowledge.

Such semantic information has a number of possible applications. One possible application is in the area of information retrieval/extraction for enhancing the search methods and for providing more precise search results. Another application is in question-answering systems, allowing users to communicate with the system in natural language (English) and translating their queries/responses into a machine-understandable representation.

We use conceptual graphs (CGs) (Sowa, 1984), a knowledge-representation formalism based on semantic networks and the existential graphs of C. S. Peirce. There is a defined mapping between a conceptual graph and a corresponding first-order logical formula, although conceptual graphs also allow for representation of temporal and non-monotonic logics, thus exceeding the expressive power of FOL.

One of the first systems for the generation of conceptual graph representation of text is described in (Sowa and Way, 1986). It uses a lexicon of canonical graphs that represent valid (possible) relations between concepts. These canonical graphs are then combined to build a conceptual graph representation of a sentence.

Veraldi at al. (1988) describe a prototype of a semantic processor for Italian sentences. It uses a lexicon of about 850 word-sense definitions, each including 10-20 surface semantic patterns (SSPs). Each SSP represents both usage information and semantic constrains and is manually acquired.

There are also systems aimed at extracting partial knowledge from texts, by either filling semantic templates (Hobbs et al., 1996) or by generation of a set of linguistic patterns for information extraction (Harabagiu and Maiorano, 2000), to name few.

The following section describes the general overview of the system, together with the documents we used to test our algorithms. Section 3 describes the semantic role identification module, Section 4 outlines the algorithm for constructing the conceptual graph representation of a sentence. The experiments that we performed are described in Section 5, while in Section 6 we draw some conclusions and outline ongoing and future work.

2 System overview

We use a two-step approach for conceptual graph representation of texts: first, by using VerbNet and WordNet, we identify the semantic roles in a sentence, and second, using these semantic roles and a set of syntactic/semantic rules we construct a conceptual graph.

The general architecture of the system is represented in Figure 1.

To apply our algorithms we use documents from two corpora in different domains. The first corpus is the freely available Reuters-21578 text categorization test collection (Reuters, 1987). The other corpus we use is the col-

lection of aviation incident reports provided by the Irish Air Accident Investigation Unit (AAIU) (2004) .

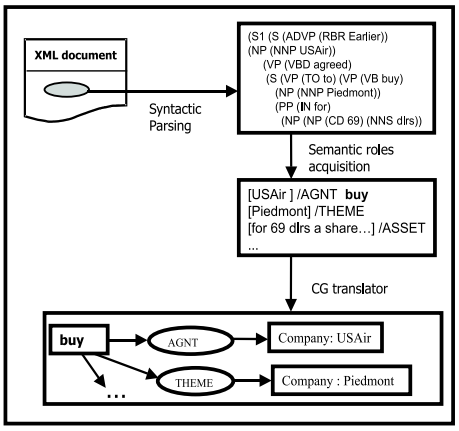

Figure 1: General architecture for the graph construction

All documents are converted to XML format and sentential boundaries are identified. The documents are then parsed using Eugene Charniak's maximum entropy inspired parser (Charniak, 2000). This probabilistic parser produces Penn tree-bank style trees and achieves 90.1% average accuracy for sentences not exceeding 40 words long and 89.5% for sentences with length under 100 words when trained and tested on the Wall Street Journal treebank.

3 Semantic role identification

The problem of automatic semantic role identification is an important part of many natural language processing systems and while recent syntactic parsers can correctly label over 95% of the constituents of a sentence, finding a representation in terms of semantic roles is still unsatisfactory.

There are number of quite different existing approaches for identifying semantic roles. The traditional parsing approaches, such as HPSG grammars and Lexical functional grammars, to a certain extent all suggest semantic relationships corresponding to the syntactic ones. They rely strongly on manually-developed grammars and lexicons, which must encode all possible realisations of the semantic roles. Developing such grammars is a time-consuming and tedious process and such systems usually work well within limited domains only.

The data-driven approach is an alternative approach, based on filling semantic templates. Applying such a model to information extraction, in AutoSlog Riloff (1993) builds a list of patterns for filling in semantic slots in a specific domain, as well as a method for automatic acquisition of case frames (Riloff and Schmelzenbach, 1998). In the domain of the Air Traveler Information System, Miller at al. (1996) apply statistical methods to compute the probability of a constituent in order to fill in a semantic slot within a semantic frame.

Gildea and Jurafsky (2000, 2002) describe a statistical approach for semantic role labelling using data collected from FrameNet. They investigate the influence of the following features for identification of a semantic role: *phrase type*, *grammatical function* (the relationship of the constituent to the rest of the sentence), *position* in the sentence, *voice* and *head word*, as well as a combination of features. They also describe a model for estimating the probability a phrase to be assigned a specific semantic role.

The approach we propose for semantic role identification uses information about each verb's behaviour, provided in VerbNet, and the WordNet taxonomy when deciding whether a phrase can be a suitable match for a semantic role.

VerbNet (Kipper et al., 2000) is a computational verb lexicon, based on Levin's verb classes (Levin, 1993), that contains syntactic and semantic information for English verbs. Each VerbNet class defines a list of *members*, a list of possible *thematic roles*, and a list of *frames (patterns)* of how these semantic roles can be realized in a sentence.

WordNet (Fellbaum, 1998) is an English lexical database containing about 120 000 entries of nouns, verbs, adjectives and adverbs, hierarchically organized in synonym groups (called *synsets*), and linked with relations, such as *hypernym*, *hyponym*, *holonym* and others.

The algorithm for semantic role identification of a sentence that we propose consists of the following three steps:

1. Firstly, for each clause in the sentence we identify the main verb and build a sentence pattern using the parse tree;

2. Secondly, for each verb in the sentence we extract a list of possible semantic frames from VerbNet, together with selectional restrictions for each semantic role;

3. Thirdly, we match the sentence pattern to each of the available semantic frames, taking into account the semantic role's constraints. As a result we are presented with a list of all possible semantic role assignments, from which we have to identify the correct one.

These steps are described in more detail in the following sub-sections.

3.1 Constructing sentence patterns for the verbs in a sentence

As mentioned earlier, during the pre-processing stage we produce a parse tree for each sentence using the Char-

niak parser. From this parse tree for each clause of the sentence we construct a sentence pattern, which is a flat parse representation that identifies the main verb and the other main categories of the clause. For example, from the parse tree for the sentence

USAir bought Piedmont for 69 dlrs cash per share

we construct the following pattern:

NP **VERB(buy)** NP PP

As a sentence can have subordinate clauses, we may have more than one syntactic pattern per sentence. Each such pattern is processed individually.

3.2 Extracting VerbNet semantic role frames

Each verb can be described in VerbNet as a member of more than one class (for example the verb *make* is listed as a member of the verb classes *dub-29.3* and *build-26.1*, each of which correspond to different verb senses), and therefore the list of its possible semantic frames is a combination of the semantic frames defined in each of the classes in which it participates (currently we do not distinguish between different verb senses and therefore do not process the WordNet sense information attached to each verb class member).

We extract all the semantic frames in a class and considers them to be possible semantic frames for each of the verbs that are members of this class. For example, for all the verbs that are members of the VerbNet class **get-13.5.1** (including the verb *buy*) we extract the semantic frames shown in Figure 2.

Agent V Theme	(1)
Agent V Theme Prep(from) Source	(2)
Agent V Theme Prep(for) Beneficiary	(3)
Agent V Beneficiary Theme	(4)
Agent V Theme Prep(for) Asset	(5)
Asset V Theme	(6)

Figure 2: Semantic frames and selectional restrictions extracted for the verbs in class **get-13.5.1**

The verb classes also define a list of selectional constraints each semantic roles should satisfy. For example, the roles defined in the VerbNet class **get-13.5.1** should satisfy the restrictions shown in Figure 3.

Some frames define additional restrictions local to the frame. In this case these restrictions are combined with the restrictions defined in the frames.

Agent[+animate OR +organization]
Theme[]
Source[+concrete]
Beneficiary[+animate OR +organization]
Asset[+currency]

Figure 3: Selectional constraints for the semantic roles defined in class **get-13.5.1**

3.3 Matching algorithm

The matching algorithm matches the sentence pattern against each of the possible semantic role frames extracted from VerbNet. We independently match the constituents before and after the verb in the sentence pattern to the semantic roles before and after the verb in the semantic role frame.

If the number of the available constituents in the sentence pattern is less than the number of the required slots in the frame, the match fails.

If there is more than one constituent available to fill a slot in a semantic frame, they are assigned priorities using heuristic rules. For example, in the cases where we have a choice of a few possible role fillers for the Agent, a higher weight is given to noun phrases, especially if they are marked as proper nouns (NNP) or contain at least one proper noun.

If, for a semantic frame, we find a constituent for each of the semantic role slots that complies with the selectional constraints, the algorithm considers this a possible match. Currently, if the algorithm returns more than one match, we manually select the best one.

3.4 Selectional constraints check

The selectional constraints check verifies if a candidate constituent for a thematic role fulfills the selectional constraints assigned to this role. For example, a common requirement for a constituent to fill the role of *Agent* is to be of type *animate* or *organization*.

The selectional constraints check is implemented using one or combination of the following techniques: hypernym relations defined in WordNet, pattern matching techniques, syntactic rules and some heuristics.

For example, the restriction *machine* is a type restriction and is fulfilled if the word represented by the constituent is a member of a synset that is a hyponym of the synset containing the word *machine*.

Other restrictions, like *infinitival* and *sentential*, are resolved only by checking the syntactic parse structure of the parse tree.

Restrictions such as *animate* and *organization* are resolved by applying a combination of the synset hierarchy in WordNet and pre-compiled lists of organization and

personal names, and if no satisfactory answer is found, using heuristics to identify if the phrase contains proper nouns.

We also check for a suitable preposition before the constituent to be matched. For example, for the frame

Agent V Topic Prep(to) Recipient

the constituent filling the semantic role of Recipient should be a prepositional phrase headed by the preposition *to* (e.g. *Bob said a few words **to Mary***).

4 Building conceptual graphs

The previous section describes the process of identifying the semantic roles of the constituents in a sentence. These roles are used to build a conceptual graph representation of the sentence by applying series of transformations, starting with more generic concepts and relations and replacing them with more specific ones.

The conceptual graph is built through the following steps:

- Step 1 – *For each of the constituents of the sentence we build a conceptual graph representation*

  Each phrase (part of the sentence) should be represented by a conceptual graph. This is done recursively by analysing the syntactical structure of the phrase.

- Step 2 – *Link all the conceptual graphs representing the constituents in a single graph*

  All the conceptual graphs built during the previous step are attached to the concept representing the verb, thus creating a conceptual graph representation for the complete sentence.

- *Step 3 – Resolve the unknown relations*

  This step attempts to identify all generic labels assigned during the previous two steps. This is done by using a list of relation correction rules.

Each of these steps are described in more detail in the following sub-sections.

4.1 Building a conceptual graph representation of a phrase

This step involves building a conceptual graph for a phrase. Our general assumption is that each lexeme in the sentence is represented using a separate concept, therefore all nouns, adjectives, adverbs and pronouns are represented using concepts, while the determiners and numbers are used as a referent of the relevant concept (thus further specifying the concept).

Here we will outline the process of building a conceptual graph for a phrase depending on the part of speech category of the phrase.

4.1.1 Noun phrases

The list of some of the most common syntactic patterns for noun phrases is shown in Table 1.

	Syntactic pattern	% AAIU	% Reuters
(1)	NP -> DT NN	20.42%	9.10%
(2)	NP -> NP PP	12.99%	14.17%
(3)	NP -> DT JJ NN	5.32%	2.49%
(4)	NP -> NN	5.18%	4.01%
(5)	NP -> NNP	4.59%	6.09%
(6)	NP -> PRP	3.57%	4.47%
(7)	NP -> NNP NNP	3.22%	2.15%
(8)	NP -> CD NNS	2.88%	1.81%
(9)	NP -> DT NN NN	2.20%	1.17%
(10)	NP -> NP SBAR	0.88%	1.29%

Table 1: A list of some of the most common syntactic patterns for noun phrases

Each of these cases is resolved individually. For example, for pattern (1) we create a concept for the NN with a referent, corresponding to the type of the determiner (an existential quantifier referent if the word marked as DT is *the*, a defined quantifier if the word is *every*, or none if the word is *a*). For pattern (3) we create concepts representing the adjective and the noun and link them by an *Attribute* relation. Pattern (10) represents phrases where the noun is further specified by the SBAR (for example, *The co-pilot, who was acting as a main pilot, landed the plane.*) For these patterns a conceptual graph is built for the SBAR and the head concept, which could be a WHNP phrase (e.g. *which* or *who*) or WHADVP (e.g. *where*) is replaced by the concept, created for the NP (also see Table 3).

4.1.2 Prepositional phrases

The conceptual graph representation of propositional phrases, similarly on the noun phrases, depends on their syntactic structure. A list of the most common syntactic patterns for prepositional phrases is shown in Table 2.

	Syntactic pattern	% AAIU	% Reuters
(1)	PP -> IN NP	77.99%	82.57%
(2)	PP -> TO NP	13.81%	8.81%

Table 2: A list of the most common syntactic patterns for prepositional phrases

The two most common patterns consist of a preposition followed by a noun phrase. For such prepositional phrases we construct a conceptual graph representing the noun phrase. We also keep track of the preposition heading the prepositional phrase, as it is used to mark the re-

lation between this phrase and the rest of relevant phrases in the sentence.

4.1.3 Subordinate clauses

The list of the most common syntactic patterns for phrases representing subordinate clauses (and marked as SBAR) is shown in Table 3.

	Syntactic pattern	% AAIU	% Reuters
(1)	SBAR -> IN S	52.76%	24.33%
(2)	SBAR -> WHNP S	18.90%	12.57%
(3)	SBAR -> WHADVP S	12.60%	2.53%
(4)	SBAR -> S	3.94%	56.34%

Table 3: A list of the most common syntactic patterns for subordinate phrases

For all these cases the embedded clause S is treated as an independent sentence, and we recursively create a conceptual graph for it. To link the resulting graph to the main graph we either use a relation with label related to the preposition marked as IN (in case (1)) or by replacing the concept representing the WHNP or the WHADVP node with the concept representing the node it refers to.

4.2 Attaching all constituents to the verb

After building separate graphs for each of the constituents, we link them together in a single conceptual graph. As each of them describe some aspect of the concept represented with the verb, we link them to that concept. Here we use the term *main node* to denote the node (concept) in the conceptual graph representing the head of the constituent. We identify the head using syntactic information about the constituent. For example, if the constituent is a noun phrase consisting of a noun phrase, followed by a prepositional phrase, its head is the head of the noun phrase and the PP is a modifier. Alternatively, if the constituent is a noun phrase that consists of an adjective followed by a noun, the noun is the head and the adjective is a modifier.

If the constituent already has a semantic role attached to it, the same relation is used when constructing the conceptual graph between the CG representing the constituent and the verb.

If the constituent does not have any semantic roles attached to it, a relation with a generic label is used. Using a generic type of relation allows us to build the structure of the CG, concentrating on the concepts involved, and to resolve the remaining relations later. If the constituent is not a propositional phrase (this includes NP, SBAR, etc.), we use a generic label *REL*.

If the constituent is a prepositional phrase (PP) headed with a proposition *prep*, we use a generic label *REL_prep*. For example, for the phrase *a flight from Dublin* we create a concept of a flight and a concept of a city, called *Dublin* and link them with a generic relation *REL_from*.

4.3 Resolving unknown relations

This is the final step in the conceptual graph construction, where we resolve the unknown (generic) relations in the conceptual graph.

We keep a database of most common syntactic realisation of relations between concepts with specific types. Figure 4 shows some of the relation correction rules we use for the documents in the AAIU corpus. The left part of the rule represents the two concepts linked with a generic relation, while the right side represents this graph after the correction. For example, the first pattern states that if in our graph there are concepts *Runway* and *Airport* linked with relation **REL_at**, we replace the relation with **Location**.

Runway **REL_at** Airport	-> Runway **Location** Airport
Flight **REL_from** Airport	-> Flight **Source** Airport
Flight **REL_from** City	-> Flight **Source** City
Flight **REL_to** Airport	-> Flight **Destination** Airport
Flight **REL_to** City	-> Flight **Destination** City
Flight **REL_for** Airport	-> Flight **Destination** Airport
Flight **REL_for** City	-> Flight **Destination** City
Land **REL_on** Runway	-> Land **Destination** Runway
Route **REL_from** City	-> Route **Source** City
Route **REL_to** City	-> Route **Destination** City

Figure 4: A sample list of relation correction rules

Building the relation correction rules database is a challenging task. Currently, the process is semi-automated by scanning the corpus for commonly occurring syntactic patterns. Such patterns are then manually evaluated and the semantic relations are identified.

Here is an example of applying a relation correction rule: for the NP *the flight from Dublin* on step 2 we create the conceptual graph

[FLIGHT:*a]->(REL_from)->[City:Dublin]

Using the correction rule 3 we substitute the relation **REL_from** with **Source** to produce the graph

[FLIGHT:*a]->(Source)->[City:Dublin]

This is an useful approach for resolving relations between nouns, as no such information is available in VerbNet.

5 Experimental results

We currently are in the process of testing and tuning our system. We have some preliminary results for the performance of the semantic role annotation module, both on Reuters news articles and AAIU reports. The tests on the Reuters documents are performed on a quarter of the

available corpus (reut2-003.sgm) and for the AAIU documents on the reports from years 1998, 1999 and 2000.

The coverage (the percentage of the verbs in the corpus that have a VerbNet description) of VerbNet for both corpora is relatively low: 66% for Reuters and 53% for the AAIU.

To evaluate the performance of the semantic role labelling algorithm we randomly selected 1% of the verbs from each corpus and manually analysed the assigned semantic roles. Our tests show that the semantic roles are correctly identified in 39% of cases in Reuters corpus and 35% of the cases in the AAIU reports, which is 59% and 66% respectively of the verbs present in VerbNet (the percentage of the correctly identified out of all that are covered by VerbNet).

We are currently extending the coverage of VerbNet by manually identifying frames present in the corpora and not included in VerbNet, which we believe should significantly increase the performance.

6 Conclusions

In this paper we described an approach for constructing conceptual graphs for English sentences, using syntactic and semantic information from VerbNet and WordNet, as well as some domain-specific knowledge. We tested the semantic role labeling algorithm on parts of Reuters corpus and on Irish Air Accident reports. The achieved accuracy is strongly influenced by the lack of VerbNet description of many verbs present in the corpora, as well as the lack of semantic frames for the verb sense.

The work on the system is ongoing and the efforts are continuing to implement a verb sense disambiguation component and to test the conceptual graph construction module.

7 Acknowledgments

This work is developed as part of the INTINN project, funded under the Enterprise Ireland Informatics Research Initiative. I would also like to thank my supervisor, John Dunnion, and the anonymous reviewers for their useful comments.

References

Air Accident Investigation Unit. 2004. Irish Air Accident Investigation Unit Reports. Available online: (http://www.aaiu.ie/).

Eugene Charniak. 2000. A Maximum-Entropy-Inspired Parser. In *Proceedings of NAACL-2000*, pages 132–139.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, May.

Daniel Gildea and Daniel Jurafsky. 2000. Automatic Labeling of Semantic Roles. In *Proceedings of 38th Annual Conference of the Association for Computational Linguistics (ACL-00)*, pages 512–520, Hong Kong, October.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245–288.

Sanda Harabagiu and Steven Maiorano. 2000. Acquisition of linguistic patterns for knowledge-based information extraction. In *Proceedings of LREC-2000*, Athens, June.

Jerry Hobbs, Douglas Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. 1996. FASTUS: A cascaded finite-state transducer for extracting information from natural-language text. In *In Finite State Devices for Natural Language Processing*, Cambridge, MA. MIT Press.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-Based Construction of a Verb Lexicon. In *Proceedings of Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 691 – 696, Austin, TX, July 30 - August 3.

Beth Levin. 1993. *English Verb Classes And Alternations: A Preliminary Investigation*. The University of Chicago Press.

Scott Miller, David Stallard, Robert Bobrow, and Richard Schwartz. 1996. A fully statistical approach to natural language interfaces. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 55–61, Santa Cruz, CA, June. Morgan Kaufmann Publishers, Inc.

Reuters. 1987. Reuters-21578 Text Categorization Collection. Available online: (http://kdd.ics.uci.edu/-databases/reuters21578/reuters21578.html).

Ellen Riloff and Mark Schmelzenbach. 1998. An Empirical Approach to Conceptual Case Frame Acquisition. In *Proceedings of the Sixth Workshop on Very Large Corpora*.

Ellen Riloff. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 811–816. AAAI Press/The MIT Press.

John F. Sowa and Eileen C. Way. 1986. Implementing a semantic interpreter using conceptual graphs. *IBM Journal of Research and Development*, 30(1):57–69, January.

John F. Sowa. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, M.

Paola Velardi, Maria Teresa Pazienza, and Mario De'Giovanetti. 1988. Conceptual graphs for the analysis and generation of sentences. *IBM Journal of Research and Development*, 32(2):251–267, March.