Constructing Conceptual Graphs using Linguistic Resources

Svetlana Hensman and John Dunnion

Intelligent Information Retrieval Group
Department of Computer Science
University College Dublin
Dublin, Ireland

ABSTRACT

With the huge number of documents becoming available in electronic form, finding relevant information in a large corpus is becoming an increasingly important, but difficult, task. We believe that semantic processing is required in order to achieve more accurate information retrieval. This paper describes a framework for the creation of semantic markup and its insertion into XML documents. We describe the semi-automatic construction of conceptual graph representations of texts using a combination of existing linguistic resources, such as VerbNet and WordNet. The system we have developed uses a two-step approach, firstly identifying the semantic rôles in a sentence, and then using these rôles, together with semi-automatically compiled domain-specific knowledge, to construct the conceptual graph representation.

Keywords: Conceptual Graphs, semantic markup, semantic rôles, WordNet, VerbNet.

1. INTRODUCTION

With the huge number of documents becoming available in electronic form, finding relevant information in a large corpus is becoming an increasingly important, but difficult, task. We believe that semantic processing is required in order to achieve more accurate information retrieval.

Manually creating semantic representation for a large corpus is a time-consuming task. The automation of this task in a general domain is not trivial, as it requires deep syntactic processing, detailed lexical resources and a great deal of domain knowledge; thus building semantic representations of documents has been one of the most challenging tasks in the area of natural language processing.

This paper describes a technique for the semi-automatic construction of conceptual graphs using a combination of linguistic resources, such as VerbNet and WordNet, together with semi-automatically compiled domain-specific knowledge. The availability of such semantic information would be useful in a number of applications. One possible application is the enhancement of search methods to provide more precise search results in the areas of information retrieval and information extraction. Another application is in question-answering systems, allowing users to communicate with the system in natural language and translating their queries and responses into a machine-understandable representation.

We use *conceptual graphs* (CGs) [1], a knowledge-representation formalism based on semantic networks and the existential graphs of CS 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 earliest systems for the generation of conceptual graph representation of text is described in [2]. 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 [3] 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 [4] or by generation of a set of linguistic patterns for information extraction [5], to name few.

The next section describes the general overview of the system and the documents we used to test our algorithms. The semantic rôle identification module is described in Section 3. 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. OVERVIEW OF APPROACH

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

To test and evaluate our algorithms we use documents from two corpora from different domains. The first corpus is the freely available Reuters-21578 text categorisation test collection [6]. The other corpus we use is the collection of aviation incident reports provided by the Irish Air Accident Investigation Unit (AAIU) [7].

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 [8]. This probabilistic parser produces Penn tree-bank style trees and achieves an average accuracy of 90.1% 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. IDENTIFYING SEMANTIC RÔLES

Automatic semantic rôle identification is an important problem in 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 rôles is still unsatisfactory.

There are number of quite different existing approaches for identifying semantic rôles. 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 heavily on manually-developed grammars and lexicons, which must encode all possible realisations of the semantic rôles. Developing such grammars is a time-consuming and tedious process and such systems usually work well only within limited domains.

An alternative approach is the data-driven approach, which is based on filling semantic templates. Applying such a model to information extraction in the AutoSlog system, Riloff [9] builds a list of patterns for filling in semantic slots in a specific domain, automatic acquiring case frames [10]. In the domain of the Air Traveler Information System, Miller at al [11] 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 [12, 13] describe a statistical approach for semantic rôle labelling using data collected from FrameNet. They investigate the influence of the following features for identification of a semantic rôle: *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 rôle.

The approach we propose for semantic rôle identification uses information about each verb's behaviour, provided in VerbNet, and the WordNet taxonomy to decide whether a phrase is a potential suitable match for a semantic rôle.

VerbNet [14] is a computational verb lexicon, based on Levin's verb classes [15], that contains syntactic and semantic information for English verbs. Each VerbNet class defines a list of *members*, a list of possible *thematic rôles*, and a list of *frames (patterns)* of how these semantic rôles can be realised in a sentence.

WordNet [16] is an English lexical database containing about 120 000 entries of nouns, verbs, adjectives and adverbs, hierarchically organised in synonym groups (called *synsets*), and linked with relations, such as *hypernym*, *hyponym*, *holonym* and others.

The algorithm that we propose for the identification of semantic rôles in a sentence 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 rôle;
3. Thirdly, we match the sentence pattern to each of the available semantic frames, taking into account the semantic rôle's constraints. As a result we are presented with a list of all possible semantic rôle assignments, from which we have to identify the correct one.

These steps are described in more detail in the following subsections.

3.1. Constructing sentence patterns for the verbs in a sentence

As mentioned above, during the pre-processing stage we produce a parse tree for each sentence using the Charniak parser. 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 rôle 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 consider 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 1.

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 1: 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 rôle should satisfy. For example, the rôles defined in the VerbNet class **get-13.5.1** should satisfy the restrictions shown in Figure 2.

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

Figure 2: Selectional constraints for the semantic rôles defined in class **get-13.5.1**.

Some frames define additional restrictions local to the frame: such restrictions are combined with the restrictions defined in the frames.

3.3. Matching algorithm

The matching algorithm matches the sentence pattern against each of the possible semantic rôle frames extracted from VerbNet. We independently match the constituents before and after the verb in the sentence pattern to the semantic rôles before and after the verb in the semantic rôle 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 rôle 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 rôle slots that complies with the selectional constraints, the algorithm considers this a possible match. Currently, if the algorithm returns more than one match, the best one is selected manually.

3.4. Selectional constraints check

The selectional constraints check verifies if a candidate constituent for a thematic rôle fulfills the selectional constraints assigned to this rôle. For example, a common requirement for a constituent to fill the rôle 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 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 rôle of *Recipient* should be a prepositional phrase headed by the preposition *to* (eg *Bob said a few words* ***to Mary***). The previous section described the process of identifying the semantic rôles of the constituents in a sentence. These rôles are used to build a conceptual graph representation of the sentence by applying a series of transformations, starting with more generic concepts and relations and replacing them with more specific ones.

The following steps summarise the conceptual graph construction process:

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

  Each phrase (part of the sentence) is 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.

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

3.5. 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.

3.5.1. Noun phrases

Some of the most common syntactic patterns for noun phrases are shown in Table 1.

NP -> NP PP	(1)
NP -> NNP (NNP ...)(or NNPS)	(2)
NP -> DT NN	(3)
NP -> NN	(4)
NP -> NNS	(5)
NP -> DT JJ NN	(6)
NP -> JJ NN	(7)
NP -> PRP	(8)
NP -> NP , SBAR ,	(9)
NP -> NP , SBAR	(10)
NP -> NP SBAR	(11)
NP -> NN CD	(12)

Figure 3: Some of the most common syntactic patterns headed by NP.

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. The final pattern, Pattern (21), 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 (eg *which* or *who*) or

	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 -> JJ NN	1.66%	1.75%
(11)	NP -> NN CD	1.51%	0.03%
(12)	NP -> NP CC NP	1.32%	1.67%
(13)	NP -> DT JJ NN NN	1.17%	0.53%
(14)	NP -> DT NNS	1.17%	1.31%
(15)	NP -> NP	1.12%	0.00%
(16)	NP -> NP VP	1.12%	1.39%
(17)	NP -> NNS	0.98%	3.39%
(18)	NP -> CD	0.93%	1.51%
(19)	NP -> PRP$ NN	0.93%	0.61%
(20)	NP -> NP NN	0.88%	1.35%
(21)	NP -> NP SBAR	0.88%	1.29%

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

WHADVP (e.g. *where*), is replaced by the concept created for the NP (see also Table 3).

3.5.2. Prepositional phrases

Similarly to noun phrases, the conceptual graph representation of a propositional phrase depends on its 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%
(3)	PP -> IN S	2.86%	2.28%

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 relation between this phrase and the other relevant phrases in the sentence.

3.5.3. Subordinate clauses

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

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.

3.6. Linking 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. Section 3 describes how to identify the semantic rôles for some of the constituents.Each of the constituents should be represented using a conceptual graph and the main node should be linked to the verb with an appropriate relation. 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 (PP), 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 rôle 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 rôles 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 prepositional 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*.

3.7. 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**.

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.

The following is an example of applying a relation correction rule. For the NP *the flight from Dublin*,we create the following conceptual graph in step 2:

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

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

	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%
(5)	SBAR -> WHPP S	2.36%	1.11%

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

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.

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

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

4. EXPERIMENTAL RESULTS

We have implemented our system and have carried out initial experiments. An overview of the system is presented in Figure 5. We are currently performing further testing and tuning of our system. We have some preliminary results for the performance of the semantic rôle annotation module, both on Reuters news articles and AAIU reports. The system has been tested on a quarter of the available corpus of Reuters documents (`reut2-003.sgm`) and on the AAIU reports for the 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 the Reuters corpus and 53% for the AAIU corpus.

To evaluate the performance of the semantic rôle labelling algorithm we randomly selected 1% of the verbs from each corpus and manually analysed the assigned semantic rôles. Our tests show that the semantic rôles 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 improve system performance.

5. CONCLUSIONS

In this paper we have described a technique for the semi-automatic construction of conceptual graphs for English texts, using syntactic and semantic information from VerbNet and WordNet, as well as some domain-specific knowledge. We have tested the semantic rôle labelling algorithm on parts of a corpus of Reuters documents and on Irish Air Accident reports. The achieved accuracy is heavily dependent on the fact that many verbs present in the corpora have no description in VerbNet, as well as the fact that there are no semantic frames for some verb senses. Work on the system is ongoing and efforts are continuing to implement a verb sense disambiguation component and to test the conceptual graph construction module.

6. ACKNOWLEDGEMENTS

The support of the Enterprise Ireland Informatics Research Initiative is gratefully acknowledged. The research was funded under grant PRP/00/INF/06.

7. REFERENCES

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

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

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

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

[5] Sanda Harabagiu and Steven Maiorano. Acquisition of linguistic patterns for knowledge-based information ex-

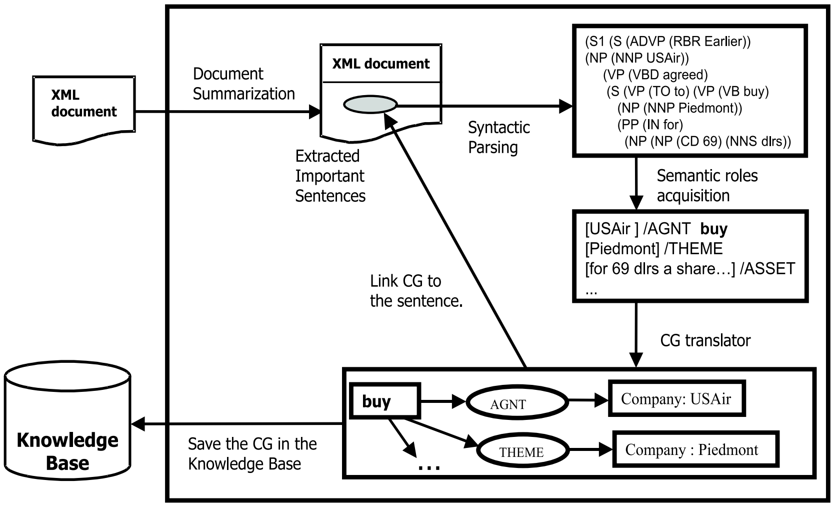

Figure 5: Overview of conceptual graph construction system.

traction. In *Proceedings of LREC 2000*, Athens, June 2000.

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

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

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

[9] Ellen Riloff. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI 1993)*, pages 811–816. AAAI Press/The MIT Press, 1993.

[10] Ellen Riloff and Mark Schmelzenbach. An Empirical Approach to Conceptual Case Frame Acquisition. In *Proceedings of the 6th Workshop on Very Large Corpora*, 1998.

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

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

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

[14] Karin Kipper, Hoa Trang Dang, and Martha Palmer. Class-Based Construction of a Verb Lexicon. In *Proceedings of 17th National Conference on Artificial Intelligence (AAAI 2000)*, pages 691–696, Austin, TX, July 30–August 3 2000.

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

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