Matching Syntactic-Semantic Graphs for Semantic Relation Assignment

Vivi Nastase[1] and **Stan Szpakowicz**[1,2]
[1] School of Information Technology and Engineering,
University of Ottawa, Ottawa, Canada
[2] Institute of Computer Science,
Polish Academy of Sciences, Warsaw, Poland
{vnastase,szpak}@site.uottawa.ca

Abstract

We present a graph-matching algorithm for semantic relation assignment. The algorithm is part of an interactive text analysis system. The system automatically extracts pairs of syntactic units from a text and assigns a semantic relation to each pair. This is an incremental learning algorithm, in which previously processed pairs and user feedback guide the process. After each assignment, the system adds to its database a syntactic-semantic graph centered on the main element of each pair of units. A graph consists of the main unit and all syntactic units with which it is syntactically connected. An edge contains information both about syntax and about semantic relations for use in further processing. Syntactic-semantic graph matching is used to produce a list of candidate assignments for 63.75% of the pairs analysed, and in 57% of situations the correct relations is one of the system's suggestions; in 19.6% of situations it suggests only the correct relation.

1 Introduction

When analysing texts, it is essential to see how elements of meaning are interconnected. This is an old idea. The first chronicled endeavour to connect text elements and organise connections between them goes back to the 5^{th} century B.C. and the work of Panini[1]. He was a grammarian who analysed Sanskrit (Misra, 1966). The idea resurfaced forcefully at several points in the more recent history of linguistic research (Tesnière, 1959; Gruber, 1965; Fillmore, 1968). Now it has the attention of many researchers in natural language processing, as shown by recent research in semantic parsing and semantic role labelling (Baker et al., 1998; Kipper et al., 2000; Carreras and Marquez, 2004; Carreras and Marquez, 2005; Atserias et al., 2001; Shi and Mihalcea, 2005).

Graph-like structures are a natural way of organising one's impressions of a text seen from the perspective of connections between its simpler constituents of varying granularity, from sections through paragraphs, sentences, clauses, phrases, words to morphemes.

In this work we pursue a well-known and often tacitly assumed line of thinking: connections at the syntactic level reflect connections at the semantic level (in other words, syntax carries meaning). Anecdotal support for this stance comes from the fact that the grammatical notion of case is the basis for semantic relations (Misra, 1966; Gruber, 1965; Fillmore, 1968). Tesnière (1959), who proposes a grouping of verb arguments into *actants* and *circumstances*, gives a set of rules to connect specific types of *actants* – for example, *agent* or *instrument* – to such grammatical elements as subject, direct object, indirect object. This idea was expanded to include nouns and their modifiers through verb nominalizations (Chomsky, 1970; Quirk et al., 1985).

We work with sentences, clauses, phrases and words, using syntactic structures generated by a parser. Our system incrementally processes a text, and extracts pairs of text units: two clauses, a verb and each of its arguments, a noun and each of its modifiers. For each pair of units, the system builds a syntactic graph surrounding the main element (main clause, head verb, head noun). It then tries to find among the previously processed instances another main element with a matching syntactic graph. If such a graph is found, then the system maps previously assigned semantic relations onto the current syntactic graph. We have a list of 47 relations that manifest themselves in compound clauses, inside a simple clause or in noun phrases. The list, a synthesis of a number of relation lists cited in the literature, has been designed to be general, domain-independent (Barker et al., 1997a).

Section 2 overviews research in semantic relation analysis. Section 3 describes the text we used in ex-

[1] The sources date his work variously between the 5^{th} and 7^{th} century.

periments, and the semantic relation list. Section 4 looks in detail at the graph-matching heuristic. Section 5 describes the experimental setting and shows how often the heuristic was used when processing the input text. We show in detail our findings about syntactic levels (how often graph matching helped assign a relation between two clauses, a verb and its arguments, or a noun and its modifier) and about the accuracy of the suggestion. Discussion and conclusions appear in Section 6.

2 Related Work

Some methods of semantic relation analysis rely on predefined templates filled with information from processed texts (Baker et al., 1998). In other methods, lexical resources are specifically tailored to meet the requirements of the domain (Rosario and Hearst, 2001) or the system (Gomez, 1998). Such systems extract information from some types of syntactic units (clauses in (Fillmore and Atkins, 1998; Gildea and Jurafsky, 2002; Hull and Gomez, 1996); noun phrases in (Hull and Gomez, 1996; Rosario et al., 2002)). Lists of semantic relations are designed to capture salient domain information.

In the Rapid Knowledge Formation Project (RKF) a support system was developed for domain experts. It helps them build complex knowledge bases by combining components: events, entities and modifiers (Clark and Porter, 1997). The system's interface facilitates the expert's task of creating and manipulating structures which represent domain concepts, and assigning them relations from a relation dictionary.

In current work on semantic relation analysis, the focus is on semantic roles – relations between verbs and their arguments. Most approaches rely on *VerbNet* (Kipper et al., 2000) and *FrameNet* (Baker et al., 1998) to provide associations between verbs and semantic roles, that are then mapped onto the current instance, as shown by the systems competing in semantic role labelling competitions (Carreras and Marquez, 2004; Carreras and Marquez, 2005) and also (Gildea and Jurafsky, 2002; Pradhan et al., 2005; Shi and Mihalcea, 2005).

These systems share two ideas which make them different from the approach presented here: they all analyse verb-argument relations, and they all use machine learning or probabilistic approaches (Pradhan et al., 2005) to assign a label to a new instance. Labelling every instance relies on the same previously encoded knowledge (see (Carreras and Marquez, 2004; Carreras and Marquez, 2005) for an overview of the systems in the semantic role labelling competitions from 2004 and 2005). Pradhan et al. (2005) combine the outputs of multiple parsers to extract reliable syntactic information, which is translated into features for a machine learning experiment in assigning semantic roles.

Our system analyses incrementally pairs of units coming from three syntactic levels – clause (CL), intra-clause (or verb-argument, IC), noun-phrase (NP). There are no training and testing data sets. Instead of using previously built resources, the system relies on previously processed examples to find the most appropriate relation for a current pair. Because the system does not rely on previously processed or annotated data, it is flexible. It allows the user to customize the process for a specific domain by choosing the syntactic units of interest and her own list of relations that best fit the domain.

It is also interesting to assess, using the current system configuration, the effect of syntactic information and incremental learning on semantic analysis. This is described in section 5.

Because of these differences in the type of data used, and in the learning approach, the results we obtain cannot be compared to previous approaches. In order to show that the system does learn, we show that the number of examples for which it provides the correct answer increases with the number of examples previously analysed.

3 Input data and semantic relations

3.1 Input data

We work with a semi-technical text on meteorological phenomena (Larrick, 1961), meant for primary school students. The text gradually introduces concepts related to precipitation, and explains them. Its nature makes it appropriate for the semantic analysis task in an incremental approach. The system will mimic the way in which a human reader accumulates knowledge and uses what was written before to process ideas introduced later in the text.

The text contains 513 sentences, with an average length of 9.13 words. There are 4686 word tokens and 969 types. The difference between the number of types (2850) and tokens (573) in the extracted pairs (which contain only open-class words) shows that the same concepts recur, as expected in a didactic text.

The syntactic structures of the input data are produced by a parser with good coverage and detailed syntactic information, DIPETT (Delisle and Szpakowicz, 1995). The parser, written in Prolog, implements a classic constituency English grammar from Quirk et al. (1985). Pairs of syntactic units connected by grammatical relations are extracted from the parse trees. A dependency parser would produce a similar output, but DIPETT also provides verb subcategorization information (such as, for example, subject-verb-object or subject-verb-object-indirect_object), which we use to select the (best) matching syntactic structures.

To find pairs, we use simple structural information. If a unit is directly embedded in another unit, we assume a subordinate relation between them; if the two units are coordinate, we assume a coordinate relation. These assumptions are safe if the parse is correct. A modifier is subordinate to its head noun, an argument to its head verb, and a clause perhaps to the main clause in the sentence.

If we conclude that two units should interact, we seek an appropriate semantic relation to describe this interaction. The system uses three heuristics to find one or more semantic relation candidates for the current pair.

1. **Word match** – the system will propose the semantic relation(s) that have previously been assigned to a pair containing the same lemmas.

2. **Syntactic graph match** – we elaborate this heuristic in Section 4.

3. **Marker** – the system uses a manually built dictionary of markers (prepositions, coordinators, subordinators) associated with the semantic relations they indicate. The dictionary contains 325 markers, and a total of 662 marker-relation associations.

If neither of the three heuristics yield results, the system will present an empty list, and expect the user to input the appropriate relation. When at least one relation is proposed, the user can *accept* a unique relation, *choose* among several options, or *supply* a new one. The system records which action took place, as well as the heuristic that generated the options presented to the user. The pair is also analysed to determine the syntactic level from which it came, to allow for a more detailed analysis of the behaviour of the system.

3.2 Semantic relations

The list of semantic relations with which we work is based on extensive literature study (Barker et al., 1997a). Three lists of relations for three syntactic levels – inter-clause, intra-clause (case) and noun-modifier relations – were next combined based on syntactic and semantic phenomena. The resulting list is the one used in the experiments we present in this paper. The relations are grouped by general similarity into 6 relation classes (H denotes the head of a base NP, M denotes the modifier).

1. CAUSAL groups relations enabling or opposing an occurrence. Examples:

   cause - H causes M: *flu virus*;

   effect - H is the effect (was caused by) M: *exam anxiety*;

   purpose - H is for M: *concert hall*;

2. CONJUNCTIVE includes relations that describe the conjunction or disjunction of occurrences (events/act/actions/states/activities), entities or attributes:

   conjunction - both H and M occur or exist (and nothing more can be said about that from the point of view of causality or temporality): *running and swimming (are good for you)*;

   disjunction - either one or both H and M occur or exist: *painting or drawing*;

3. PARTICIPANT groups relations between an occurrence and its participants or circumstances. Examples:

   agent - M performs H: *student protest*;

   object - M is acted upon by H: *metal separator*;

   beneficiary - M benefits from H: *student discount*;

4. SPATIAL groups relations that place an occurrence at an absolute or relative point in space. Examples:

   direction - H is directed towards M: *outgoing mail*;

   location - H is the location of M: *home town*;

   location at - H is located at M: *desert storm*;

5. TEMPORAL groups relations that place an occurrence at an absolute or relative point in time. Examples:

   frequency - H occurs every time M occurs: *weekly game*;

   time at - H occurs when M occurs: *morning coffee*;

   time through - H existed while M existed: *2-hour trip*;

6. QUALITY groups the remaining relations between a verb or noun and its arguments. Examples:

   manner - H occurs as indicated by M: *stylish writing*;

material - H is made of M: *brick house*;
measure - M is a measure of H: *heavy rock*;

There is no consensus in the literature on a list of semantic relations that would work in all situations. This is, no doubt, because a general list of relations such as the one we use would not be appropriate for the semantic analysis of texts in a specific domain, such as for example medical texts. All the relations in the list we use were necessary, and sufficient, for the analysis of the input text.

4 Syntactic-semantic graph-matching

Our system begins operation with a minimum of manually encoded knowledge, and accumulates information as it processes the text. This design idea was adopted from TANKA (Barker et al., 1997b). The only manually encoded knowledge is a dictionary of markers (subordinators, coordinators, prepositions). This resource does not affect the syntactic-semantic graph-matching heuristic.

Because the system gradually accumulates knowledge as it goes through the input text, it uses a form of memory-based learning to make predictions about the semantic relation that fits the current pair. The type of knowledge that it accumulates consists of previously analysed pairs, together with the semantic relation assigned, and a syntactic-semantic graph centered on each word in a sentence which appears as the main element in a processed pair.

To process a pair P not encountered previously, the system builds a graph centered on the main element (often the head) of P. This idea was inspired by Delisle et al. (1993), who used a list of arguments surrounding the main verb together with the verb's subcategorization information and previously processed examples to analyse semantic roles (case relations). In recent approaches, syntactic information is translated into features which, together with information from *FrameNet, WordNet* or *VerbNet*, will be used with ML tools to make predictions for each example in the test set (Carreras and Marquez, 2004; Carreras and Marquez, 2005).

Our system builds a (simple) graph surrounding a head word (which may be a verb – representing the predicate of a sentence, or representing a clause – or noun), and matches it with previously analysed examples.

A graph $G(w)$ centered on word w consists of the following: a node for w; a set of nodes for each of the words w_i in the sentence with which w is connected by a grammatical relation (including situations when w is a modifier/argument); edges that connect w with each w_i, tagged with grammatical relation GR (such as *subject, object, complement*) and connective information Con (prepositions, coordinators, subordinators, or *nil*). The nodes also contain part-of-speech information for the corresponding word, and other information from the parser (such as subcategorization structure for the verb, if it is available).

Graph matching starts with the central node, and continues with edge matching. If $G(w)$ is the graph centered on word w whose pairs are currently being processed, the system selects from the collection of previously stored graphs, a set of graphs $\{G(w_i)\}$, which satisfy the following conditions:

- The central nodes match. The matching is guided by a set of contraints. We choose the graphs centered on the nodes that satisfy the most constraints, presented here in the order of their importance:
  - w and w_i must have the same part of speech.
  - w and w_i have the same syntactic properties. If w and w_i are verbs, they must have the same subcategorization structure.
  - w and w_i are the same lemma. We emphasize that a graph centered on a different lemma, but with the same subcategorization structure is preferred to a graph with the same lemma, but a different subcategorization structure.
- The edge representing the word pair to which we want to assign a semantic relation has a match in $G(w_i)$. From all graphs that comply with this constraint, the ones that have the lowest distance – corresponding to the highest matching score – are chosen. The graphs are matched edge by edge. Two edges match if the grammatical relation and the connectives match. Figure 1 shows the formula that computes the distance between two graphs. We note that edge matching uses only edge information – grammatical and connective information. Using the node information as is (lemmas and their part of speech) is too restrictive. We are looking into using word similarity as a solution of node matching.

If no matching graph has been found, the system searches for a simpler match, in which the current word pair is matched against previously processed pairs, using the same formula as for edge distance, and preferring the pairs that have the same modifier.

This algorithm will retrieve the set of graphs $\{G(w_i)\}$, which give the same score when matched

Definition of a graph centered on w:

$$G(w) = \{w_i, edge(w,w_i) \text{ or } edge(w_i,w) | w_i \text{appears in sent. S, and is connected to } w\}$$

$$edge(w,w_i) = \{GR_i, Con_i\} \ ; \quad \begin{array}{l} GR_i \in \{subject, object, complement, ...\} \\ Con_i \in \{at, in, on, with, for, ...\} \end{array}$$

Distance metric between two graphs:

$$dist(G(w_1), G(w_2)) = \sum_{k=1}^{N} d(edge_{1k}, edge_{2k}); \qquad edge_{ik} \in G(w_i), N \text{ is the number of edges in } G(w_i)$$

$$\begin{aligned} d(edge_{1k}, edge_{2k}) &= d(\{GR_{1k}, Con_{1k}\}, \{GR_{2k}, Con_{2k}\}) \\ &= d_1(GR_{1k}, GR_{2k}) + d_1(Con_{1k}, Con_{2k}) \end{aligned}$$

$$d_1(x,y) = \left\{ \begin{array}{lll} 0 & : & x = y \\ 1 & : & x \neq y \end{array} \right.$$

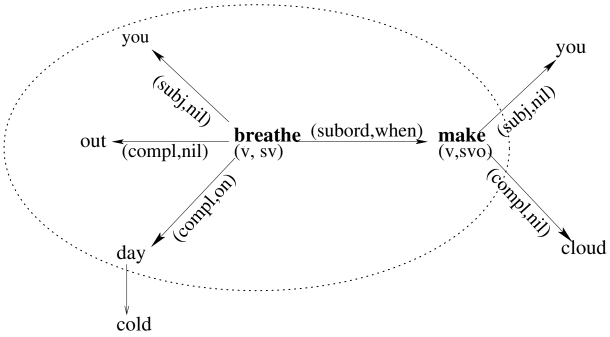
Figure 1: Distance between two graphs

with the current graph. The set of possible semantic relations presented to the user consists of the semantic relation on the edge of each $G(w_i)$ that matches the edge (of the current graph) corresponding to the word pair which we are analysing.

To the sentence:

When you breathe out on a cold day, you make a cloud.

corresponds the following syntactic graph:



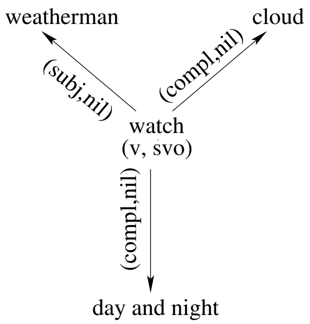
When we focus on the graph centered on a specific word, such as *breathe*, we look only at the node corresponding to the word *breathe*, and the nodes adjacent to it.

To process a pair $P = (w_H, w_M)$, the system first builds $G(w_H)$, and then searches through previously stored graphs for those which have the same center w_H, or have the same part of speech as w_H assigned to its central node. For each graph found, we compute a distance that gives a measure of the match between the two graphs. The best match will have the smallest distance.

For example, for the sentence:

Weathermen watch the clouds day and night.

the system builds the following network centered on the predicate *watch*[2]:



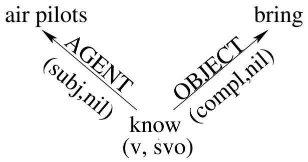
The system locates among previously stored networks those centered around verbs[3]. For the sentence above, the system uses the following graph,

[2] The *nil* value on the edges means that no preposition or other connective explicitly links the two words or the corresponding syntactic structures.

[3] If more detailed information is available, the system will choose only networks associated with verbs that have the same subcategorisation pattern (svo, svoi and so on).

built from the immediately preceding sentence in the text:

Air pilots know that clouds can bring rain, hail, sleet and snow.



According to the metric, the networks match and the pairs **(watch, weatherman)** and **(know, air pilots)** match, so the semantic relation for the pair **(know, air pilots)** is proposed as a possible relation for pair **(watch, weatherman)** .

5 Experiments

The system processes the 513 sentences interactively. It begins by running the DIPETT parser. Next, it extracts syntactic units (clauses, phrases, words) and pairs them up according to the information in the parse tree. Each unit is represented by its head word. Next, the system checks if the same pair of word lemmas has already been processed, to propose the same relation(s) to the user as options. If not, the system builds a graph centered on the head word, and proceeds with the matching on previously encountered instances, as described in section 4. When a set of candidates has been found, the system goes through a dialogue with the user.

The system generated 2020 pairs from the 513 sentences. The experiment was run in 5 interactive sessions of approximately 3 hours each. The total net processing time was 6 hours, 42 minutes and 52 seconds[4]. While it would have been instructive to run the system several times with different users, it was not feasible. The experiment was run once, with two cooperating users. They were instructed on the set of semantic relations, and told how the system works. They discussed the semantic relation assignment and, once agreed, compared the system's suggestion with their decision.

DIPETT did not produce a complete parse for all sentences. When a complete parse (correct or incorrect) was not possible, DIPETT produced fragmentary parses. The semantic analyser extracted units even from tree fragments, although sometimes the fragments were too small to accommodate any pairs. Of the 513 input sentences, 441 had a parse tree that allowed the system to extract pairs.

# of analyzed examples	1475		
level statistics	CL	IC	NP
	64	978	433
user actions	accept	choose	supply
	459	393	623
avg. # of suggestions	2.81		
graph-matching usage	933		
level/action statistics	CL	IC	NP
accept 183 (19.61%)	9	141	33
choose 349 (37.41%)	23	314	12
supply 401 (42.98%)	27	316	58

Table 1: Summary of semantic analysis

Of 2020 pairs generated, the users discarded 545 in the dialogue step. An example of an erroneous pair comes from the sentence:

Tiny clouds drift across like feathers on parade.

The semantic analyser produced the pair *(drift,parade)*, because of a parsing error: *parade* was parsed as a complement of *drift*, instead of a post-modifier for *feathers*. The correct pairing *(feather,parade)* is missing, because it cannot be inferred from the parse tree.

Table 1 gives a summary of the processing statistics. We observe that graph-matching was used to process a clear majority of the total pairs extracted – 63.25% (933/1475) , leaving the remaining 36.75% for the other two heuristics and for cases where no suggestion could be made. In 57.02% of the situations when graph-matching was used, the system's suggestion contained the correct answer (user's action was either *accept* or *choose*), and in 19.61% of the situations a single correct semantic relation was proposed (user action was *accept*).

When the system presents multiple suggestions to the user, including the correct one, the average number of suggestions is 3.75. The small number of suggestions shows that the system does not simply add to the list relations that it has previously encountered, but it learns from past experience and graph-matching helps it make good selections. Figure 2 plots the difference between the number of examples for which the system gives the correct answer (possibly among other suggestions) and the number of examples when the user must supply the correct relation, from the first example processed until the end of the experiment. We observe a steady increase in the number of correctly processed examples.

Our system does not differentiate between syntactic levels, but based on the structures of the syntactic units in each pair we can decide which syntactic level it pertains to. For a more in-depth analysis, we have separated the results for each syntactic level,

[4] The time difference accounts for system processing times, and user interaction for other steps of the analysis.

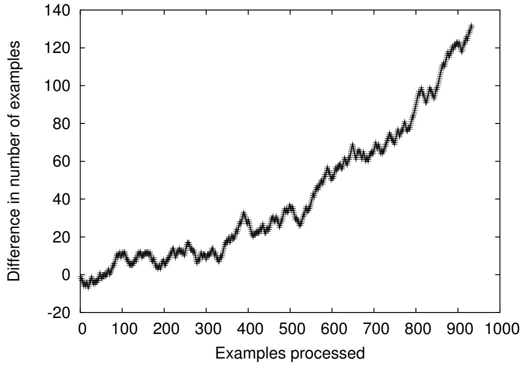


Figure 2: Difference between the number of situations in which the user *accepts* or *chooses* from the system's suggestions, and when it must *supply* the correct relation

and present them for comparison in Figure 3.

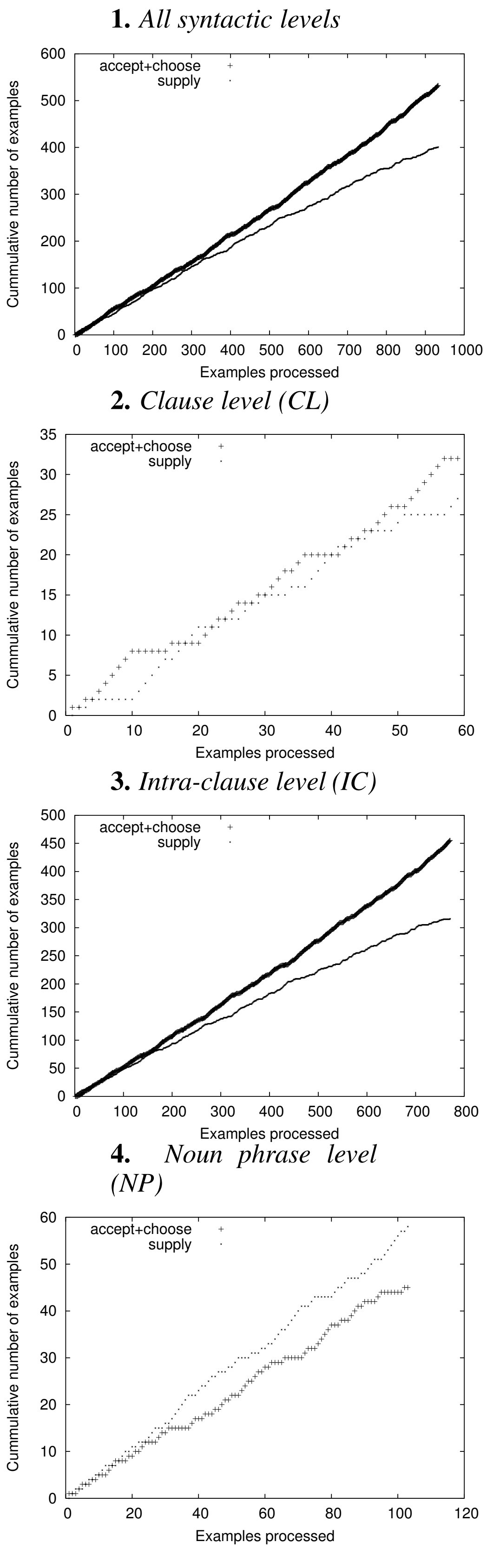
We observe that the intra-clause level – verbs and their arguments – makes the best use of graph-matching, with the curve showing the cumulative number of situations in which the system makes correct predictions becoming steeper as more text is processed. At the same time, the curve that plots the cumulative number of cases in which the user has to supply a correct answer begins to level off. As expected, at the noun-phrase level where the syntactic structures are very simple, often consisting of only the noun and its modifier (without even a connective), the graph-matching algorithm does not help as much. At the inter-clause level the heuristic helps, as shown by the marginally higher curve for cumulative *accept/choose* user actions, compared to *supply* actions.

6 Conclusions

We have shown through the results gathered from an interactive and incremental text processing system that syntactic-semantic graph-matching can be used with good results for semantic analysis of texts. The graph-matching heuristic clearly dominates other heuristics used, and it learns to make better predictions as more examples accumulate.

Graph-matching is most useful for assigning semantic relations between verbs and their arguments, but it also gives good results for inter-clause relations. At the noun-phrase level, we could only tackle noun-modifier pairs that exhibit a modicum of syntactic structure – a connective. For base NPs there is practically nothing that syntactic information can bring to the semantic analysis process.

The graph-matching process could be improved by bringing into play freely available lexical re-



1. *All syntactic levels*

2. *Clause level (CL)*

3. *Intra-clause level (IC)*

4. *Noun phrase level (NP)*

Figure 3: Graph-matching for different syntactic levels

sources. For now, the actual words in the graph nodes are not used at all. We could use *WordNet* to compute word similarities, to select closer matching graphs. *VerbNet* or *FrameNet* information could help choose graphs centered on verbs with similar syntactic behaviour, as captured by Levin's verb groups (Levin, 1993) which are the basis of *VerbNet*.

References

Jordi Atserias, L. Padró, and German Rigau. 2001. Integrating multiple knowledge sources for robust semantic parsing. In *Proceedings of RANLP - 2001*, Tsigov Czark, Bulgaria.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING-ACL*, pages 86–90, Montreal, Canada.

Ken Barker, Terry Copeck, Sylvain Delisle, and Stan Szpakowicz. 1997a. Systematic construction of a versatile case system. *Journal of Natural Language Engineering*, 3(4):279–315.

Ken Barker, Sylvain Delisle, and Stan Szpakowicz. 1997b. Test-driving TANKA: Evaluating a semi-automatic system of text analysis for knowledge acquisition. In *Proceedings of CAI 1997*, pages 60–71, Vancouver, BC, Canada.

Xavier Carreras and Lluis Marquez, editors. 2004. *Introduction to the CoNLL-2004 Shared Task: Semantic Role Labelling*. Boston, MA, USA.

Xavier Carreras and Lluis Marquez, editors. 2005. *Introduction to the CoNLL-2005 Shared Task: Semantic Role Labelling*. Ann Arbour, MI, USA.

Noam Chomsky. 1970. Remarks on nominalizations. In Roderick Jacobs and Peter Rosenbaum, editors, *Readings in English Transformational Grammar*, pages 184–221. Ginn and Co., Waltham, MA, USA.

Peter Clark and Bruce Porter. 1997. Building concept reprezentations from reusable components. In *AAAI*, pages 367–376, Providence, Rhode Island.

Sylvain Delisle and Stan Szpakowicz. 1995. Realistic parsing: Practical solutions of difficult problems. In *PACLING*, Brisbane, Queensland, Australia.

Sylvain Delisle, Terry Copeck, Stan Szpakowicz, and Ken Barker. 1993. Pattern matching for case analysis: A computational definition of closeness. In *ICCI*, pages 310–315, Sudbury, ON, Canada.

Charles Fillmore and Beryl T. Atkins. 1998. FrameNet and lexicographic relevance. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, Granada, Spain.

Charles Fillmore. 1968. The case for case. In Emmond Bach and Robert T. Harms, editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart and Winston.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Fernando Gomez. 1998. A representation of complex events and processes for the acquisition of knowledge from text. *Kowledge-Based Systems*, 10(4):237–251.

Jeffrey Gruber. 1965. *Studies in Lexical Relations*. Ph.D. thesis, MIT, Cambridge, MA. Reprinted in Jeffrey Gruber. 1976. *Lexical Structures in Syntax and Semantics*. Part I. North-Holland Publishing Company, Amsterdam.

Richard D. Hull and Fernando Gomez. 1996. Semantic interpretation of nominalizations. In *13th National Conference on Artificial Intelligence*, pages 1062–1068, Portland, Oregon, USA.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *AAAI/IAAI*, pages 691–696.

Nancy Larrick. 1961. *Junior Science Book of Rain, Hail, Sleet and Snow*. Garrard Publishing Company, Champaign, Illinois.

Beth Levin. 1993. *English Verb Classes and Alternations*. University of Chicago Press.

Vidya Niwas Misra. 1966. *The Descriptive Technique of Panini*. Mouton, The Hague.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Daniel Jurafsky. 2005. Semantic role labelling using different syntactic views. In *Proceedings of ACL 2005*, pages 581–588, Ann Arbour, MI, USA.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, London and New York.

Barbara Rosario and Marti Hearst. 2001. Classifying the semantic relations in noun-compounds via a domain specific hierarchy. In *EMNLP*, pages 82–90, Pittsburg, PA, USA.

Barbara Rosario, Marti Hearst, and Charles Fillmore. 2002. The descent of hierarchy and selection in relational semantics. In *ACL*, Philadelphia, PA, USA.

Lei Shi and Rada Mihalcea. 2005. Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing. In *Proceedings of CICLing 2005*, pages 100–111, Mexico City, Mexico.

Lucien Tesnière. 1959. *Éléments de syntaxe structurale*. C. Klincksieck, Paris.