Learning generic semantic roles

RIK DE BUSSER rik.debusser@law.kuleuven.ac.be
Interdisciplinary Centre for Law & IT, K.U.Leuven, Tiensestraat 41, 3000 Leuven, Belgium
T: +32 16 32 51 77 – F: +32 16 32 54 38

MARIE-FRANCINE MOENS marie-france.moens@law.kuleuven.ac.be
Interdisciplinary Centre for Law & IT, K.U.Leuven, Tiensestraat 41, 3000 Leuven, Belgium
T: +32 16 32 53 83 – F: +32 16 32 54 38

Abstract. Generic event detection is one of the main steps that natural language analysis still has to take to come to a stage in which computers really start to understand what a text is about. One approach to this task analyzes individual clauses into patterns of semantic roles that represent fundamental functional categories that reflect how humans conceptualize reality. We have designed a method for learning these domain-independent semantic roles from an annotated corpus by combining notions of frame theory and of systemic-functional grammar and simplifying the process to a pattern classification process. We have conducted a number of experiments which indicate that a regular mapping exists between superficial lexical, morphological and positional features in the surface structure of a text and its functional-semantic deep structure and that we can therefore semantically classify phrases using only superficial language analysis.

Keywords: event detection, information extraction, semantic classification, frame theory, functional grammar, syntactic-semantic interface

1. Introduction

The interest in text understanding is as old as artificial intelligence (AI) itself. Traditionally, the relationship between linguistic expressions and event semantics is often encoded in knowledge structures such as semantic frames (Minsky, 1975; Schank, 1972). In information extraction (IE) systems, these expressions are mapped into the frames to extract event information from text. Such an approach has two main obstacles: manually building large sets of semantic frames or annotating a corpus to build them automatically is an extremely labor-intensive task; and it is hard to construct a coherent and consistent frame set because of the inherent ambiguity of natural language semantics. In this article we present an approach that incorporates notions of systemic-functional grammar into frame semantics. Since the former is a generic theoretical-linguistic framework for understanding text, our approach will largely avoid the two obstacles. Systemic-functional grammar starts from the observation that humans perceive reality through a mediating set of conceptual categories that are reflected in the lexico-grammatical constructs of a language (Halliday & Matthiessen, 1999). Instead of the ad-hoc semantic roles of traditional IE, it allows us to identify domain-independent categories that represent actions and states in the real world. We have designed a method for detecting these categories in sentences by interpreting the frame building process as a pattern classification task and we have proved that it can be used to detect generic semantic roles in a reliable way. Eventually, this will allow text understanding and IE to break free from domain-dependence and give them a basic understanding of the events expressed in a text. In the next section, we will define our problem and justify our research. Section 3 describes our methodology from the point-of-view of linguistics and machine learning. In section 4, we will describe our experiments and their results; section 5 is a discussion of the results and of the application potential of our technique. Before the conclusion, we give a concise history of semantic roles and mention some related research.

2. Problem definition and goal of the research

It is one of the greatest frustrations of AI, natural language processing (NLP) and related domains that computers so far have stubbornly failed to acquire the ability to relate utterances in a text to some kind of conceptual model of the world, notwithstanding the research efforts that were made during the last decades. Apart from the computational complexity of the task, there are two main obstacles for putting real understanding into computers. A first problem is that determining the exact meaning of an utterance requires a considerable amount of knowledge about the text and the world in general, making that any sophisticated form of natural language understanding requires the existence of a comprehensive world model (or domain model), a linguistic model and conversational models to interpret the function of an utterance in the flow of reasoning.

A second problem is that – unlike in areas such as morphology or syntax – there is no agreement at all about which relationships between language and meaning exactly exists and how they should be formally implemented in a computational-linguistic framework. Consider for example the following sentence.

(1) The prime minister dissolved parliament.

There will be little objection to a part-of-speech tagger marking the word 'parliament' as a noun, since that is about the only word class that can be assigned to the word, and similarly, a syntactic parser will almost always classify 'parliament' as a noun phrase that is dominated by the verb phrase with main verb 'dissolved'. However, a uniform semantic classification does not exist, neither on the level of individual words, nor on any larger scale.

A distinction between 'semantics of truth' and 'semantics of understanding' (i.e. conceptual semantics) was first made by Fillmore (1985). *Truth semantics* deals with the sufficient and necessary conditions for making valid judgments about event descriptions, i.e., it primarily deals with the validity of statements that refer to real-world events and is only indirectly concerned about their verity. A major advantage of a truth-semantic framework is that it is inherently a formalized model for describing linguistic statements of events and is therefore able to perform operations on them and to keep track of their truth-conditional consistency. Unfortunately, the rigidity of a logical framework does not always correspond to human notions of what is true or false and in itself truth semantics might be able to say whether a statement is valid or not, but it cannot tell anything about what exactly the statement is about. In many ways, a *semantics of understanding* or *conceptual semantics* is complementary to truth semantics, since it describes the relationship between linguistic entities and their conceptualization of events in the real world. Conceptual frameworks do not specifically aim at assessing the validity of an expression, but rather describe how language users would interpret it in terms of an internalized cognitive world model. From all theories, we will here only consider frame theory and systemic-functional grammar.

In *frame theory* (Minsky, 1975; Winograd, 1975) event types are encoded as semantic frames, each consisting of a number of attribute-value pairs called *frame elements*. An expression of an event is represented by selecting an appropriate frame and instantiating each frame element by mapping segments of the expression into the value slots in accordance with constraints on those values. For instance, we could define a semantic frame for the meaning of 'dissolve' as it is used for the disbandment of institutional bodies. There are likely to be two obligatory participants involved: one that performs the act of dissolving and another that is being dissolved (we will disregard adverbials such as time, place or manner). As a constraint, both participants have to be expressed as a noun. The result is a frame structure such as Figure 1a, which can be instantiated with chunks of sentence (1) as in Figure 1b.

```
Frame: dissolve-institution

    event :     <value1>
    dissolver : <value2> [noun]
    dissolved : <value3> [noun]
```

(a)

```
Frame: dissolve-institution

    event :     'dissolved'
    dissolver:  'the prime minister'
    dissolved:  'parliament'
```

(b)

Figure 1. Frame representation (1) (a) and its instantiation by sentence (1) (b).

Although a frame-semantic description of complex events did not turn out to be as easy as it initially seemed, the idea of using frame-like structures for detecting individual events has been a constant in the AI and NLP communities up to this day. Despite its obvious potential, frame theory has some major drawbacks. Any realistic system needs a huge amount of semantic frames; manually building them is an extremely labor-intensive task; and designing an internally consistent frame set is far from straightforward. As a result, systems relying on frame theory have often been developed in an ad-hoc fashion for very restricted domains, leading to high maintenance and portability costs. The introduction of a generic semantic framework can solve these problems by providing internal consistency and a domain-independent classification.

Systemic-functional grammar (Halliday & Matthiessen, 1999) is such a framework. Its most basic assumption is that humans perceive reality through a mediating set of fundamental conceptual categories, which are covert but are mirrored in the lexico-grammatical constructs of a language. Since these categories reflect that human observers conceive the world as series of actions and states, a linguistic expression can be analyzed in terms of the event that it describes, the things and persons participating in it, and its setting. In the center of a functional-semantic pattern is a process of a particular type, which consists of a number of semantic roles: the *process role* itself describes an event in the real world; *participant roles* represent real-world entities participating in the event; and *circumstantial roles* describe the general setting. Using the functional proto-roles Agent and Patient, we can analyze sentence (1) as in Figure 2.

Process type: Action		
Participant: Agent	**Process: Action**	**Participant: Patient**
The prime minister	dissolved	parliament

Figure 2. Functional analysis of sentence (1) with proto-roles of Agent, Action and Patient.

A crucial difference between frame theory and systemic-functional grammar is that the latter organizes linguistic expressions into a coherent classification of domain- and language independent conceptual categories instead of ad-hoc frame elements. With the former being able to translate events descriptions into a computer-readable format and the latter connecting language and meaning, it seems like an obvious step on the way to automatic event detection to combine both into one framework.

Because generic semantic roles are only indirectly realized in language, they cannot be easily detected. Our main research aim is to discover whether and to what extent it is possible to automatically identify them in the phrases of a text based on their superficial properties (lexical, morphosyntactic and positional information). On a more theoretical level this corresponds to the search for a regular mapping between the lexical, morphological and syntactic surface structure of language and the functional-semantic deep structure. On an implementational level our goal is to build a semantic tagger or classifier that assigns generic roles to free text, using as little external resources as possible. We specifically aim at discovering roles that are domain-independent and we will learn all mapping rules from an annotated corpus with standard machine learning techniques.

3. Methodology

3.1. Linguistic framework

Frame theory gives us a formalized framework for representing event descriptions, but it largely fails to generalize across individual domains and to produce a consistent theory of functional event semantics. Systemic-functional grammar has given us such a theory at a rather informal level, making it not directly useful for actual implementations. We have tried to integrate both into one formal frame-theoretical model. The process types of systemic-functional grammar correspond to uninstantiated generic event frames; functional roles to frame elements; and assigning functional roles to mapping them into the appropriate frame elements. Automatically determining a mapping between the surface structure of linguistic utterances and the semantic deep structure simply consists of learning which lexical and morphosyntactic constraints apply on the frame elements.

The systemic-functional theory we use was developed by Michael Halliday (Halliday, 1994). He distinguishes five major process types. Actions and states in the external world are modeled in *material processes*; inner experience is expressed in *mental processes*; *relational processes* classify or identify entities or events relative to each other or to a abstract category such as color, size, etc.; *behavioral processes* express typical forms of physiological behavior; *verbal processes* model what one says or thinks; and *existential processes* state the existence of something.

(2) The lion chased the tourist. (material process)
(3) She felt distinctly unhappy about his decision. (mental process)
(4) His dog is his best friend. (relational process)

(5) He screamed with fear. (behavioral process)
(6) The woman answered: "I really have no idea." (verbal process)
(7) There are birds without feathers. (existential process)

Each of these process types has the potential to generate a number of functional role patterns, which all consist of the process role itself, a set of participants, and some optional circumstantial roles. For sentence (1), this results in the analysis in Figure 3.

Participant: Material: Actor	**Process: Material**	**Participant: Material: Goal**
The prime minister	dissolved	parliament

Figure 3. Hallidayan analysis of sentence 1.

The functional analysis reveals that 'dissolved' expresses an action in the real world, that this action has a participant, the Actor, who does the dissolving and a participant that is dissolved, the Goal. An important aspect of our classification is that every semantic role has been defined as a list of inclusion relationships (e.g. participant – participant of a material process – actor of a material process), which makes it easy to add additional layers of detail. The process and participant roles in Halliday's systemic-functional grammar all behave in a more or less regular way because they are a manifestation of the transitivity system of a language, i.e., the system that expresses the flow of individual events as an interaction between regular alternations in the order of the phrases in a clause and the lexical and morphological properties of the words in these phrases. For instance, the disbandment of an organization like a parliament can only be expressed in a limited number of constructions and with certain lexical items, as sentences (8) to (11) illustrate (an asterisk indicates invalid constructions).

(8) The prime minister dissolved parliament.
(9) Parliament was dissolved by the prime minister.
(10) Parliament was dissolved.
(11) * Parliament dissolved.

It would be very hard to detect these alternations or constraints on their behavior directly, especially because they involve several linguistic subsystems, but fortunately systemic-functional grammar presupposes a realizational chain in language generation: the semantic deep structure of language is realized in its surface structure through a number of consecutive and predictable linguistic levels (cf. Halliday & Matthiessen, 1999, p. 4).[1] Going up the chain, it can be assumed that a constellation of surface features will reflect the semantic deep structure and although a one-to-one correspondence between these antipodal linguistic strata will certainly not exist when features are considered in isolation, it might be possible to trace certain combinations of features back to their deep-semantic sources.

Our current research aims at discovering whether and how we can automatically detect the transition from a sentence such as 'The prime minister dissolved parliament' to a generic semantic role pattern such as the one in Figure 3, taking into account only superficial linguistic features. At present, we limited the functional patterns to the ones who have a verbal group for a process role. We also excluded a number of circumstantial roles, restricting our semantic classes to the ones listed in Table 1.

3.2. Machine learning framework

From a machine learning point-of-view, we consider the detection of semantic roles as a pattern classification task based on the contextual features of their corresponding phrases in a clause. For each verb, we will learn a set of discriminative surface features that allows us to relate semantic roles to particular instances of that verb in a text. The classification has the following steps. First, the textual data used in training and testing is preprocessed. Sentence boundaries are detected, the corpus is tagged with a part-of-speech tagger and rudimentary noun and verb phrase

[1] This idea was already implicitly present in the work of Panini (see Kiparsky, 2000).

boundaries are indicated. Then, the corpus is manually labeled with semantic roles and a number of training and test sets are generated. For each semantic role, contextual features are extracted and the classifier is trained on the labeled data. In the testing phase, the semantic class with the highest probability is assigned to unlabeled data. In an optional phase, the classification is refined by using information on valid co-occurrences of individual semantic roles in semantic role patterns.

Table 1. List of semantic roles used in the experiments.

- Material Process
  - Actor in a Material Process
  - Goal in a Material Process
  - Recipient in Material Process
  - Client in Material Process
  - Entity Range in a Material Process
  - Process Range in a Material Process
- Mental Process of Perception
- Mental Process of Affection
- Mental Process of Cognition
  - Senser in a Mental Process
  - Phenomenon in a Mental Process
- Behavioral Process
  - Behaver in a Behavioral Process
- Verbal Process
  - Sayer in a Verbal Process
  - Verbiage in a Verbal Process
  - Target in a Verbal Process
  - Receiver in a Verbal Process
- Existential Process
  - Existent in an Existential Process

- Intensive Attributive Relational Process
- Possessive Attributive Relational Process
- Circumstantial Attributive Relational Process
  - Carrier in an Attributive Relational Process
  - Attribute in an Attributive Relational Process
  - Beneficiary in an Attributive Relational Process
- Intensive Identifying Relational Process
- Possessive Identifying Relational Process
- Circumstantial Identifying Relational Process
  - Identified in an Identifying Relational Process
  - Identifier in an Identifying Relational Process
- Circumstance of Extent: Distance
- Circumstance of Extent: Duration
- Circumstance of Extent: Frequency
- Circumstance of Location: Place
- Circumstance of Location: Time
- Circumstance of Motion: Place towards
- Circumstance of Motion: Place from
- Circumstance of Motion: Time

In our selection of contextual features for the pattern classification task, we did not extract a limited number of features based on any prior knowledge about their fitness, but rather selected as many potentially relevant features as possible, relying on the classifiers to discriminate good from bad features. These features include word stem, word class (part-of-speech) and general word class, the position relative to the dominating process role, the properties of that process role, and a composite feature that simulates subject-object distinctions. Figure 4 presents all the features corresponding to the first phrase in Figure 3.

'The prime minister'

1: word stem:	'the', 'prime', 'minister'
2: word class:	determiner, base adjective, common noun
3: general word class:	determiner, adjective, noun
4: relative location:	before process role
5: absolute location:	1 position before process role
6: word stem of process:	'dissolve'
7: word class of process:	past simple verb
8: general word class process:	verb
9: subject/object simulation:	head of role = noun & head of process = verb & role before process

Figure 4. Surface features for 'The prime minister' in Figure 3.

Features 1-3 and 6-8 are extracted for the head of each relevant phrase and for the first and last token of their right and left context, as shown in Figure 5. For each training and test example these features are transformed into a fixed-length vector, with nominal and numeric values being translated to binary features for some of the classifiers (e.g., naïve Bayes).

left context				HEAD	**right context**				
the	huge	American	**company**	**car**	**of**	our	new	executive	**director**

Figure 5. Examples of selection of tokens per phrase (selected tokens in bold).

We trained the resulting data sets on a number of classifiers, all of which have been successfully applied to NLP in the past. Since we do not focus on the development of novel pattern classification algorithms, we mostly used existing software.

k-Nearest neighbor classification does not learn by generalizing examples and separating positive from negative ones to identify a class, but simply compares the feature vector of a new case with the vectors of existing examples that are stored in a database. Assuming that similar instances have similar classifications, it identifies for each new case the *k* closest examples for which the similarity exceeds a certain threshold and collects its class. Geometrically, no general form exists to draw a boundary between the classes, because the nearest neighbor can produce any arbitrarily complex surface to separate classes based only on the configuration of the sample points and their similarity or distance metric to one another. We have used a *k*-nearest neighbor classifier that uses a simple function for computing the distance between the test and training examples (Aha, Kibler & Albert, 1991). The distance function takes the square root of the sum of the distances between the individual features x_i and y_i defined as $f(x_i, y_i) = (x_i - y_i)^2$ for numeric valued attributes and $f(x_i, y_i) = (x_i \neq y_i)$ for Boolean and symbolic-valued attributes. In our implementation, multiple similar examples all vote for a class to be assigned with an equal weight. Nearest neighbor classification performs very well when good precedent examples with accurate, non-noise features are available for training. Also, the avoidance of generalization or abstraction has some advantages in NLP, since it is often impossible to use general rules without considering the exceptions (Daelemans, 1999).

It is possible to learn the probability distribution of a feature or a combination of features from a set of training examples, e.g. with maximum likelihood estimation. In *naïve Bayes classification* (Mitchell, 1997, 154 ff.) computations are simplified by assuming that the probabilities of the occurrence of features are independent. The probability estimate of an individual feature is based on the co-occurrence of a class and a feature in the training corpus. The probability of class membership of a new instance is computed as the product of the probabilities of class membership of the features of this instance (possibly corrected by an a priori probability of the class distribution in the training examples) and the *k* classes with the highest probability are assigned to a test example. Because maximum likelihood is only estimated from a limited sample of training data that is possibly noisy, our naïve Bayes classifier corrects the maximum likelihood estimation by assuming a normal distribution. Although naïve Bayes classification is a simple approach to pattern classification, it is often able to compete with more sophisticated (and computationally more expensive) classifiers. It has been successfully applied in word sense disambiguation problems (e.g., Pedersen, 2000).

Maximum entropy modeling (MEM) computes the probability distribution with maximum entropy that satisfies the constraints set by the training examples (Berger, Della Pietra & Della Pietra, 1996; Ratnaparkhi, 1997). This distribution has an exponential form:

$$P(o|c) = \frac{1}{Z}\prod_{j=1}^{k} \alpha_j^{f_j(c,o)}$$

where *o* refers to the outcome or class; *c* to the context; *Z* is a normalizing constant; and *k* the number of contextual features. Normally, it uses an iterative procedure such as generalized iterative scaling (GIS) to estimate the model parameter α_j of each feature function $f_j(c, o)$. The latter is a binary function that is true when the context expressed by the function is true and zero otherwise. For a new instance, membership for each class is computed and the *k* classes with the highest probability are assigned. In contrast to naïve Bayes, the model takes into account possible dependencies between features and MEM also accurately discriminates relevant from irrelevant features during training, thus relieving the implementer from selecting accurate features manually. In NLP, MEM has been successfully used for parsing (e.g., Charniak, 2000), word sense disambiguation (e.g., Chao & Dyer, 2002) and named entity recognition (e.g., Chieu & Ng, 2002).

Decision-based learners learn classifying expressions (usually rules or trees presented in a logical formalism) from a set of examples. Most of these algorithms do not exhaustively search through all possible combinations of features or feature relations to find the rules that best discriminate the positive from the negative examples for a class, but since our feature set is limited, the effect of this greediness is not very large. A new case is classified by applying the appropriate classifying expression and assigning the corresponding class. Our propositional learner is an improved version of the C4.5 algorithm (Quinlan, 1993), in which a top-down decision tree is built by iteratively selecting the feature with the largest information gain or expected reduction in entropy, which is computed by using it to partition the example set according to the target classification, and adding it as a node in the decision tree. At each iteration, a descendant of the node is created for each possible value of the selected attribute and the training examples are sorted appropriately. The advantage of decision-based learning is that the expressions that are learned can be interpreted and altered manually and that feature dependencies are naturally modeled. Decision rules have been used in semantic classification and information extraction (e.g., Hayes, 1992).

Currently, *support vector machines* (SVM) are successfully used for pattern classification (Vapnik, 1995). An SVM finds the hyperplane in the *n*-dimensional feature space that best separates positive and negative examples in the training set with maximum margins. Unlike with decision-based learning, the number of features does here not influence the results and SVMs have the ability to model exceptions by allowing a limited amount of training errors. Since both are useful properties for NLP, SVMs are increasingly used in, e.g., named entity recognition (Isozaki & Kazawa, 2002).

Semantic roles can only occur in a semantic role pattern in a limited number of possible combinations and the co-occurrence of two or more semantic roles is always dependent. Information about these dependencies can be manually acquired and implemented as symbolic knowledge (or learned from a corpus) and can further improve semantic classification. This could be done by combining the probability ranking of each semantic role provided by the classifier with the knowledge of possible combinations of roles to compute the probability of each valid combination of roles (e.g., as average of the individual probabilities of the roles). We have integrated this approach in the MEM classifier, but it could be added to any classifier that provides a probabilistic ranking.

4. Experiments

4.1. Experimental setup

Our method for learning the correlation between superficial linguistic features and individual semantic roles relies on an annotated corpus. For all trainings and tests we have used a subset of the new Reuters Corpus (*Reuters Corpus. Volume 1: English Language, 1996-08-20 to 1997-08-19*).[2] Since corpus annotation is a labor-intensive task and we had limited resources, we selected a relatively small number of verbs for evaluation that was representative for the body of verbs in English. In order to avoid bias, we simply picked an initial set of 37 verbs based on their relative frequency in the British National Corpus (Leech, Rayson & Wilson, 2001), eliminating all verbs that could be used as an auxiliary. We selected four verbs at the top of the BNC frequency list (*say*, *make*, *go* and *see*); 10 verbs in the middle of the list with a relative frequency of 31 occurrences per million words, one of which had to be discarded because it did not occur in our training corpus; and 23 verbs with a relative frequency of 16 per million words, six of which had to be discarded for the same reason. When semantic roles were learned for each verb separately, another two medium-frequency verbs and eight low-frequency verbs were rejected because their data sets did not contain at least 10 separate semantic roles and they could therefore not be tested with ten-fold cross-validation. In experiments with aggregated example sets (see Table 2), these verbs have been included. Below is an overview of all verbs that were used in the experiments.

High-frequency verbs (+1500 occurrences per 1m words in BNC):
make (1420), go (617), say (950), see (1036)
Medium-frequency verbs (31 occurrences per 1m words in BNC):
award (17), capture (114), deserve (21), distribute (37), enhance (18), sweep (16), tackle (16)
semantic roles < 10: bury (3), doubt (9)
Low-frequency verbs (16 occurrences per 1m words in BNC):
aid (21), decrease (23), dissolve (36), endorse (12), import (50), schedule (49), ship (28)

[2] Available at *http://about.reuters.com/researchandstandards/corpus/*; we used documents 77363newsML to 80419newsML.

semantic roles < 10: coincide (6), confess (3), contract (3), creep (7), dispose (3), formulate (6), impress (7), sponsor (9), tempt (3), spare (3)

In the preprocessing phase the entire corpus is cleaned; a part-of-speech tagger (with LTPOS; see Mikheev, 1997) assigns word classes to each token; noun phrases, verb phrases and prepositional phrases are detected (with LTChunk; see Mikheev, 2000); the word stem is derived for each token with a dictionary. A job student semantically annotated the resulting corpus. He was instructed to use Halliday (1994) as a manual and apart from some very general remarks concerning methodological issues, he was free to interpret the rules derived from it as he thought fit. For each verb except for *make*, he checked the annotations for consistency. In a one-month period the student managed to annotate 1450 semantic role patterns, corresponding to 4543 individual roles. The annotated texts were translated to an intermediate XML-format from which the feature vectors that are used in the experiments were extracted.

In preliminary experiments, we used C4.5 (Quinlan, 1993) for learning semantic role patterns and we evaluated the output manually (De Busser, Angheluta & Moens 2002). For most of the experiments in this article, we used version 3.3.6 of the Waikato Environment for Knowledge Analysis (Witten & Frank, 2000).[3] The classifiers borrowed from this package are: a *k*-nearest neighbor model with *k* = 1 (B); a naïve Bayes algorithm (C); J4.8, a Java implementation of C4.5 (E); and a support vector machine with a linear kernel function (F). All these tests learned individual semantic roles without combining them into semantic role patterns and were performed using ten iterations of ten-fold cross-validation (i.e., 100 runs) with randomized example order in the training and test set. We also used an open-source maximum-entropy classifier[4], which was evaluated with single ten-fold cross-validation. In a separate experiment, we implemented an optional refinement module of this classifier that concatenated the roles into semantic role patterns. In all experiments except for one (Table 2), roles were learned for each verb separately.

For each verb-per-verb training and for trainings on the aggregated training set, an average accuracy is calculated. The verb-specific classifications are also evaluated by macro-averaging (A_1) and micro-averaging (A_2), the latter giving a more realistic measurement when the test set reliably reflects a realistic distribution of verb occurrences. A_1 is the sum of the average accuracies of individual verbs averaged over the number of verbs; A_2 is averaged over the number of instances and is calculated as

$$A2 = \frac{\sum_{1}^{v}(i_v \times A_v)}{\sum_{1}^{v} i_v}$$

(where v = the number of verbs, i_v= the number of training instances per verb and A_v its average accuracy). All classifiers are compared to a baseline, which builds a one-level binary decision tree for each data set.

4.2. Results

In our preliminary experiments (De Busser et al., 2002), we learned semantic role patterns as a concatenation of case roles, following the assumption that the interaction between individual roles is best captured in the learning process by treating all semantic roles in a single pattern as one homogenous block. These experiments were especially valuable for acquiring better insights into which features play a role in the classification process. In our present experiments we learn semantic roles based on information about their internal structure and their relation to the process role. This corresponds to the idea in functional linguistics that the process role is central to the semantic pattern and all other roles are defined in relation to it. Individual roles are recombined into patterns in a separate phase.

In a first experiment, we compared a number of different classifiers (Table 1). All score well above the baseline, the SVM being slightly better than the others. On the whole A_1 is a little higher than A_2, mainly because the former overemphasizes low and medium frequency verbs and these are likely to perform better than high-frequency verbs since their semantic behavior is less varied. There is a relatively large difference in accuracy between separate verbs:

[3] Available at *http://www.cs.waikato.ac.nz/ml/weka/*

[4] Available at *http://maxent.sourceforce.net/index.html*

even when considering 'award' and 'endorse' as outliers, the results of the SVM range between 70.47 and 97.98 % (with a baseline of 43.88 % for A_2). As far as we could detect, this difference cannot be straightforwardly correlated to any single factor in the classification process and it is likely that it largely depends on the semantic complexity of individual verbs. The verbs 'award' and 'endorse' are the only two that really perform exceptionally badly, but this is caused by a lack of training examples (the decision trees that J4.8 constructs for both verbs are well-formed and intuitively correct from a linguistic point-of-view).

Table 2. Comparison of classifiers trained on individual verbs.

		A	B	C	D	E	F
	No. Inst.	**Baseline**	**1-nearest neighbor**	**Naïve Bayes**	**Maximum entropy***	**J4.8**	**SVM**
aid	21	44.3333	76.0000	88.0000	80.0000	76.0000	85.5000
award	17	13.5000	56.0000	47.5000	55.0000	53.5000	53.0000
capture	114	57.9167	90.0606	85.6288	91.2879	91.4773	89.0985
decrease	23	72.3333	63.1667	77.1667	73.3333	63.1667	83.6667
deserve	21	51.6667	96.0000	100.0000	95.0000	95.5000	86.6667
dissolve	36	58.1667	94.0000	95.0000	94.1667	94.5000	95.5833
distribute	37	48.5000	84.1667	81.7500	85.8333	84.1667	82.3333
endorse	12	33.0000	34.5000	59.0000	35.0000	34.5000	51.0000
enhance	18	33.0000	90.0000	95.0000	90.0000	90.0000	95.0000
go	617	50.3133	73.4130	69.5843	78.4321	74.4823	78.6134
import	50	47.4000	80.0000	92.6000	88.0000	79.6000	91.0000
make	1420	28.9437	59.7606	58.3451	67.8873	62.5423	70.4718
say	950	63.3684	97.5895	96.7789	98.4211	97.6737	97.9789
schedule	49	63.1500	63.2500	71.8000	73.5000	67.2500	74.0500
see	1036	39.1903	68.2036	68.1972	73.9320	72.8757	74.1612
ship	28	64.5000	71.8333	68.6667	76.6667	71.8333	72.0000
sweep	16	37.5000	72.0000	69.0000	60.0000	75.0000	75.0000
tackle	16	25.0000	89.5000	95.0000	80.0000	89.5000	95.0000
A_1		46.2101	75.4608	78.8575	77.5811	78.7720	80.5624
A_2		43.8752	73.6433	72.8053	78.6841	75.8166	79.6267

*****: Ten-fold cross-validation without iterations.

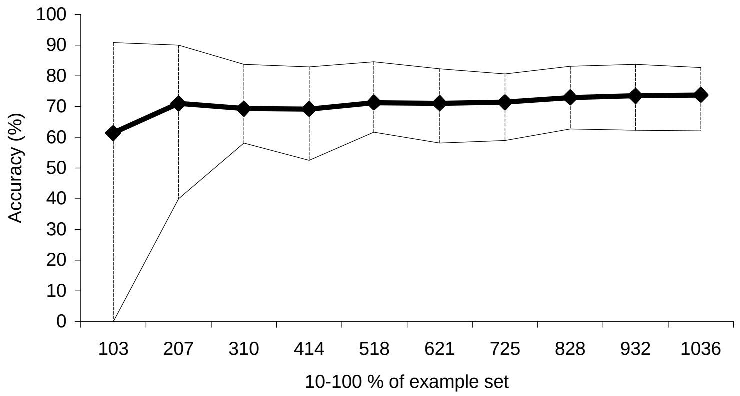

Figure 6. Influence of the size of the example set on accuracy for the verb 'see'.

We have tested whether our general setup causes overfitting on large training sets, training (F) for subsets of the data set of the verb 'see', with all settings identical to previous experiments (Figure 6). We randomized the example order and produced data sets from 10 % to 100 % of the originally size with a 10 % interval. The X-axis indicates the number of examples in the data set; the line plotted in bold is the accuracy averaged over 10 iterations of 10-fold cross-validation; the bottom line the minimum value occurring in the 100 folds for each data set; and the top line the maximum value for each data set. As expected, the accuracy increases and stabilizes around 74 % at 310 training examples (± 100 semantic role patterns) with the minimum and maximum values converging at a 20 % distance. No overfitting occurs when more training examples are added.

Table 3. Comparison of classifiers trained on aggregated example set.

		A	B	C	D	E	F
	No. Inst.	**Baseline**	**1-nearest neighbor**	**Naïve Bayes**	**Maximum entropy***	**J4.8**	**SVM**
Accuracy	4543	21.0434	70.1365	67.7063	76.8000	N/A	N/A

*: Ten-fold cross-validation without iterations.

In a third experiment, we aggregated the example sets of individual verbs and thus created one large set containing all verbs combined (see Table 3). All other parameters are identical to the first experiment. Instead of learning which surface features are indicative of the semantic roles that accompany individual verbs, this aggregated set learns similar functional behavior *across* verbs. This comparative training could be used as a classification measure complementary to the one above and to learn semantic roles for low-frequency verbs that have enough related verbs in the training set. The accuracy of all classifiers on the aggregated set dropped quite a bit compared to A_1, but these results are to be expected given the fact the complexity of the classification task increased (e.g., the average number of values per feature rose from 6.42 for A_1 to 10.06). For (E) and (F), no results could be generated due to excessive complexity of the classification.

In a fourth experiment, we identified the influence of disabling different features in the classification process. For all datasets of individual verbs, we constructed 10 subsets in which particular features are disabled (FC1-FC10). All datasets were trained on (F) with all settings identical to previous experiments. It should be noted that all subsets still have a feature configuration that makes sense from a linguistic point-of-view and is therefore discriminative at least to a minimal extent (e.g., we did never disabled all grammatical or positional features).

Table 4. Influence of disabling features.

	Feature configuration (FC)	**A_1**	**A_2**
1	Original dataset	80.5624	79.6267
2	All features of the head of role and process, relative location and subject/object simulation	81.5595	76.6298
3	Previous + absolute location	82.1555	77.0137
4	All features except for those left of the head of role and process	80.1620	79.5500
5	All features except for those right of the head of role and process	80.7182	77.1836
6	All features except for relative location	79.3843	79.2572
7	All features except for absolute location	80.4101	79.5826
8	All features except for word class	81.0504	78.7390
9	All features except for the general word class	80.3650	80.0087
10	All features except for the stem	76.2375	71.8741

Results remain quite stable across the different subsets. Only when all lexical features are disabled (FC10), accuracy drops considerably (with 4.32 and 7.75 % for A_1 and A_2 respectively). As expected the SVM is quite robust with regard to noisy and superfluous features.

In a fifth rather limited experiment, we attempted to improve semantic role detection by using knowledge of valid combinations of semantic roles in semantic role patterns as they might appear in sentences. These were manually acquired via knowledge of systemic-functional theory. The most probable valid combination of roles was computed for each clause of the test set and role assignment was corrected accordingly. We could improve the accuracy of all verbs except for 'schedule' and boost the A_2 average to 84.16 %, which indicates that the co-occurrence of semantic roles in a role pattern is dependent. In the future we would like to better define the exact nature of this dependency with more elaborate experiments.

4.3. Discussion

Our experiments prove that semantic roles are learnable from superficial linguistic features and that our methods could be implemented in an operational system. All classifiers performed well above the baseline and the results are quite consistent across classifiers. On average, the SVM with linear kernel performed somewhat better than its immediate followers. There is a fairly large difference in accuracy between separate verbs, but this is a natural consequence of the fact that all verbs have their own distinct semantic behavior and complexity.

There appears to be an upper bound to learning semantic roles that is relatively remote from the ideal of a perfect score and for which it is quite difficult to pin down an exact reason. We have manually checked all decision trees that were generated by J4.8 for overfitting and feature imbalances, but all trees that were learned corresponded to basic linguistic intuitions. Even for verbs with small example sets, J4.8 usually constructed trees that first used positional and grammatical features and only in the leave nodes resorted to lexical features (and even then only quite sparsely), which corresponds to how an average human would decide which functional role should be assigned to a particular phrase. Although it is possible that we simply need to extract more superficial linguistic features, it is difficult for us to imagine what exactly those features could be and how we could extract them from texts without using complex natural language analysis.

An important insufficiency is that we do not use any contextual features for learning individual semantic roles. Our last experiment indicates that adding information about semantic role co-occurrences will improve accuracy and in a semantic tagger this information could be added directly to the training process. Some informal experiments indicate that accuracy might improve up to 10 % in that case, but it was impossible to design simple and objective evaluation measures for this setup. A last and insurmountable barrier will always prevent semantic classification from being 100 % accurate: natural language semantics are inherently ambiguous. Sometimes, it is difficult to assign a single correct semantic role to a particular phrase because of the ambiguity of a verb or because an assignment relies on textual or world contexts, or it is even impossible because it depends on the interpretation of individual readers.

To a certain extent our experimental setup is an abstraction of the situation in which a semantic tagger would operate. Phrase boundary detection was only performed semi-automatically, since we only had access to a basic phrase chunker. In addition, by avoiding errors in the phrase boundary detection phase, we were able to identify the performance of the classification process itself more accurately and to design simple and objective evaluation measures. This makes that the accuracies in our tests are higher than they would have been in an operational system. Another simplification is that we did not yet include some circumstantial elements in semantic classification. A disadvantage of all systems that learn from annotated corpora is that it is extremely labor-intensive to annotate a corpus. In our case, the annotator needed one month to annotate 1450 semantic role patterns (or 4543 individual semantic roles). This is not as bad as it sounds. In a conservative estimate, an annotator could tag roughly 15 to 20 high-frequency verbs per person month, covering 14% to 16% of verb tokens of the British National Corpus in just one month. Moreover, one of the main reasons to use generic instead of domain-specific semantic roles is that they are reusable and one only needs to annotate a corpus once, no matter which domain it will be applied to.

A number of improvements are still necessary to scale the system up to a realistic size. We have done little work on fine-tuning classifiers (e.g., using polynomial and RBF kernels in the SVM) and we have to implement a more advanced version of the semantic role pattern combination module. In future experiments, we will also test whether augmenting the results of the verb-specific sets with those of the aggregated example set (Table 3) improves overall accuracy. More verbs need to be annotated (an average of 300-350 annotated semantic patterns per verb seems reasonable); more semantic classes need to be added; we will need to develop fixed annotation standards; and in the preprocessing phase, it will be necessary to introduce full phrase detection.

The semantic classification of clauses in terms of conceptual categories has many applications in information retrieval, information extraction and other NLP tasks, especially when only a general understanding of the event structure of a text is needed. Good examples are *question answering (QA) systems*,[5] which generate a precise answer to a natural language question (instead of giving a list of relevant documents) by extracting it from text databases. Current technologies mainly use traditional word-based information retrieval techniques to extract sentences with a potential answer from texts. The candidate answer sentence with the largest lexical and syntactic similarity is then used to generate the output answer. The introduction of semantic roles would allow us to better identify the question type and its corresponding candidate answer types and would bring event semantics in the selection process. In a similar way, developers of *search engines* might want to find related texts based on the events that they express rather than the words that they contain. In both search paradigms, information about semantic roles allows the user to zoom in or out on the content. In relaxed matching, similar semantic roles could be retrieved without matching lexical items; for very restricted matching, information could be retrieved both based on word matching and semantic classification.

Semantic role classification could also help in (semi-)automatically constructing *thesauri*, *semantic networks* and *ontologies* (see e.g. Bateman, 1990). In many knowledge-intensive applications, knowledge of functional patterns would be a great asset to break away from pure inclusion relationships and to establish links between different word classes (e.g., verbs and nouns). One could also use semantic roles for discovering synonymy and other relationships in large corpora. An important application area of semantic classifications is *information extraction*, in which extraction patterns are built to detect very specific information in text (e.g., specific events or dates). Unlike the ad-hoc extraction patterns that were used in previous systems and were only applicable in very restricted domains, our generic classification allows for a general understanding of states and events as they are expressed in general domain texts. *Full text understanding* will require huge amounts of domain and world knowledge and generic functional classification makes no pretension of providing such information. Nevertheless, it will be an indispensable first step in any sophisticated form of language understanding.

5. Related research

Semantic roles were introduced in the 6th century B.C. by the Indian grammarian Panini in his Ashtadhyayi (Vasu, 1980; for an introduction see Kiparsky, 2000). Apart from the fact that he was the first to design a comprehensive formalized grammar of a language (i.e. Sanskrit), it is important for us that he assumed that every sentence expressed a particular action and its participants. His chain of language generation therefore starts with the assignment of generic semantic roles (the *karakas*) to linguistic expressions.

The notion of semantic roles disappeared for the next 2500 years and was only unearthed in the 1960s in Fillmore's famous article on case grammar (Fillmore, 1968). His most fundamental argument is that the notion of case (i.e., the inflection of nouns, pronouns and adjectives as an expression of the function they have in a clause) is not so much connected to the lexicon, morphology or syntax but consists of a set of covert semantic roles. It is realized in the surface structure by a set of language-dependent transformation rules and as a consequence there has to be a regular mapping between the semantic deep structure and its surface realization (case markers, word order, grammatical roles, etc.). A few years later, Schank (1972) introduced a similar notion in the field of cognitive sciences. In his conceptual dependency theory, sentences are the surface realization of conceptual categories that are connected to generic action types (so-called *ACTs*). These conceptual representations are combined into multi-event frames that express the course of events for a stereotyped situation (Schank & Abelson, 1977). Unfortunately, Schank's conceptual categories were rather arbitrary and did not always correspond to how humans really conceptualized events.

In 1975, Marvin Minsky introduced the notion of frames into artificial intelligence (Minsky, 1975), a frame being a data-structure that contains attribute-value slots and represents a stereotyped state. From that time on, domain-

[5] See the TREC conferences: *http://trec.nist.gov/*

dependent frames slowly replaced the original linguistic idea of generic semantic roles. Detection of these frames was considered to be a pattern classification task, in which the classification patterns were manually drafted (e.g., DeJong, 1977; Hobbs et al., 1996; Harabagiu & Maiorano, 2000), learned from labeled examples (e.g., Riloff & Schelzenbach, 1998; Soderland, 1999; Craven et al., 2000) or partly trained from unlabelled examples (Riloff, 1996). Systems were developed for very limited subject domains and could hardly be ported to other domains, although information-rich approaches to semantic parsing which rely on huge databases of formalized world knowledge (e.g. Hahn, 1989) have been proposed in an attempt to deal with these limitations.

Recently, there have been a number of attempts to detect generic semantic roles in text. Most related to our research is the work of Gildea & Jurafsky (2002) and Fleischman, Kwon & Hovy (2003), who rely on domain-independent roles as defined in the FrameNet case frame dictionary (Johnson et al., 2001). They take into account positional, grammatical and lexical features, the main difference with our work being that both use full syntactic parsing. When training on 18 abstract semantic roles with manually assigned phrase boundaries, Gildea and Jurafsky obtained an average accuracy of 82.1 %, which is consistent with our results. For more specific semantic roles, their results are slightly lower. Their classifier learns various probability distributions of combined semantic features and assigns the most probably role to a new instance by interpolated summation of the individual distributions that apply for the new instance. We have increased our basic accuracy by adding knowledge about valid combinations of roles to the system.

In linguistics, semantic roles gave birth to diverse strands of functional grammar (e.g., Dik, 1979; Halliday, 1994; Givon, 2001), which started from the hypothesis that language arose as an expression of how humans conceived reality and therefore strongly focused on the process-semantic analysis of linguistic phenomena. Most functionalist approaches explicitly postulate the existence of a regular (though not necessarily isomorphic) mapping between the semantic stratum and the linguistic surface. By using a semantic classification scheme based on systemic-functional grammar our experiments have demonstrated the existence of this mapping.

6. Conclusion

In the research presented in this article, we have developed a method for learning individual functional-semantic roles from an annotated corpus by reducing the problem to a pattern classification task. In doing this, we have shown that there exists a regular mapping between the linguistic surface organization of a text and its hidden semantic deep structure, as has been postulated in functional linguistics. We avoid the randomness and domain-dependence of previous approaches by implementing a generic linguistic framework and cut down on development time by using a domain-independent, reusable corpus. Although there are a number of limitations to learning semantic roles and experiments were conducted in a restricted research setting, there is no indication that this would hinder our techniques from being implemented in a functioning system. Being able to unambiguously analyze the event structure of a text, numerous applications will benefit from using generic semantic roles. We therefore hope that our research is at least a small step step forward on the road to a future in which computers will slowly learn to understand what humans write and say.

Acknowledgements

This research was funded by the Institute for the Promotion of Innovation by Science and Technology in Flanders. We thank Roxana Angheluta and Rudradeb Mitra for their programming help and building the annotation tool and Koen Plevoets for his enthusiasm in a work as boring as annotating a corpus.

References

Aha, D.W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning* 6 :1, 37-66.

Bateman, J. (1990). Upper modeling: a general organization of knowledge for natural language processing. In *Proceedings of the 5th International Workshop on Natural Language Generation*. Pittsburgh, PA.

Berger, A., Della Pietra, S.A., & Della Pietra, V.J. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22, 39-71.

Chao, G., & Dyer, M. (2002). Maximum entropy models for word sense disambiguation. In *COLING 2002. Proceedings of the 19th International Conference on Computational Linguistics* (pp. 155-161). New Brunswick, NJ: ACL.

Chieu, H.L., & Ng, H.T. (2002). Named entity recognition: A maximum entropy approach using global information. In *COLING 2002. Proceedings of the 19th International Conference on Computational Linguistics* (pp. 160-167). New Brunswick, NJ: ACL.

Charniak, E. (2000). A Maximum-Entropy-Inspired Parser. In *Proceedings of NAACL-2000* (pp. 132-139). New Brunswick, NJ: ACL.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (2000). Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118:1-2, 69–113.

Daelemans, W. (1999). Machine learning approaches. In H. van Halteren (Ed.), *Syntactic Wordclass Tagging*. Dordrecht: Kluwer Academic Publishers.

De Busser, R., Angheluta, R. & Moens, M.-F. (2002). Semantic case role detection for information extraction. In *COLING 2002 - Proceedings of the 19th International Conference on Computational Linguistics* (pp. 1198-1202). New Brunswick: ACL.

DeJong , G. (1977). Skimming newspaper stories by computer. In R. Reddy (ed.), *Proceedings of the 5th International Joint Conference on Artificial Intelligence* (p. 16). Cambridge, MA: William Kaufman.

Dik, S. (1979). *Functional Grammar*. Amsterdam: North-Holland.

Fillmore, C. (1985). Frames and the semantics of understanding. *Quaderni di Semantica*, 6:2, 222-254.

Fillmore, C. J. (1968). The case for case. In E. Bach, & R.T. Harms (Eds.), *Universals in Linguistic Theory*. New York: Holt, Rinehard and Winston.

Fleischman, M., Kwon, N., & Hovy, E. (2003). Maximum entropy models for FrameNet classification. In *Empirical Methods in Natural Language Processing*. New Brunswick, NJ: ACL.

Gildea, D., & Jurafsky, D. (2002). Automatic labelling of semantic roles. *Computational Linguistics*, 28:3, 245-288.

Givon, T. (2001). *Syntax. An Introduction*. Amsterdam: John Benjamins Publishing Company.

Hahn, U. (1989). Making understanders out of parsers: semantically driven parsing as a key concept for realistic text understanding applications. *International Journal of Intelligent Systems*, 4:3, 345-339.

Halliday, M.A.K., & Matthiessen, C. (1999). *Construing Experience Through Meaning. A Language-based Approach to Cognition*. London: Cassell.

Halliday, M.A.K. (1994). *An Introduction to Functional Grammar*. London: Arnold.

Harabagiu, S.M., & Maiorano, S. (2000). Acquisition of linguistic patterns for knowledge-based information extraction. In *Proceedings of LREC-2000*. Athens, Greece.

Hayes, P.J. (1992). Intelligent high-volume text processing using shallow, domain-specific techniques. In P.S. Jacobs (Ed.), *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*. Hillsdale: Lawrence Erlbaum.

Hobbs, J.H., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., & Tyson, M. (1996). FASTUS: A cascaded finite-state transducer for extracting information from natural-language text. In E. Roche, & Y. Schabes (Ed.), *Finite State Devices for Natural Language Processing*. Cambridge, MA: MIT Press.

Isozaki, H., & Kazawa, H. (2002). Efficient support vector classifiers for named entity recognition. In *Coling 2002. Proceedings of the 19th International Conference on Computational Linguistics* (pp. 390-396). San Fransciso, CA: Morgan Kaufman.

Johnson, C.R., Fillmore, C.J., Wood, E.J., Ruppenhofer, J., Urban, M., Petruck, M.R.L., & Baker, C.F. (2001). *The FrameNet Project: Tools for Lexicon Building*. URL: http://www.icsi.berkeley.edu/~framenet/book/book.html.

Kiparsky, P. (2000). *On the Architecture of Panini's Grammar*. Three lectures delivered at the Hyderabad Conference on the architecture of grammar, Jan. 2002, and at UCLA, March 2002.

Leech, G., Rayson, P., & Wilson, A. (2001). *Word Frequencies in Written and Spoken English: Based on the British National Corpus*. London: Longman.

Mikheev, A. (1997). Part-of-speech guessing rules: Learning and evaluation. *Computational Linguistics*, 23:3, 405-423.

Mikheev, A. (2000). Document centered approach to text normalization. In N.J. Belkin, P. Ingwersen, & M. Leong (eds.), *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 136-143). Athens, Greece: ACM.

Minsky, M. (1975). A framework for representing knowledge. In P.H. Winston (Ed.), *The Psychology of Computer Vision*. New York: McGraw-Hill.

Mitchell, T.M. (1997). *Machine Learning*. Boston, MA: McGraw-Hill.

Pedersen, T. (2000). A simple approach to building ensembles of naïve Bayesian classifiers for word sense disambiguation. In *Proceedings of the First Meeting of the NAACL-00, Seattle, WA, May, 1-3 2000.*

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Ratnaparkhi, A. (1997). *A Simple Introduction to Maximum Entropy Models for Natural Language Processing*. Technical Report 97-08. IRCS, Philadelphia, PA: University of Pennsylvania.

Riloff, E., & Schelzenbach, M. (1998). An empirical approach to conceptual case frame acquisition. In *Proceedings of the Sixth Workshop on Very large Corpora*. Montreal, Canada.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (pp. 104-1049). Menlo Park, CA: AAAI.

Schank, R.C., & Abelson, R.P. (1977). *Scripts, Plans, Goals and Understanding. An Inquiry into Human Knowledge Structures*. Hillsdale, NJ: Erlbaum.

Schank, R.C. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3, 552-631.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34:1, 233-272.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer.

Vasu, S.C. (1980). *The Ashtadyayi of Panini*. Delhi: Motilal Banarsidass.

Winograd, T. (1975). Frame representations and the declarative/procedural controversy. In D. Bobrow & A. Collins (Eds.), *Representation and Understanding. Studies in Cognitive Science*. New York: Academis Press.

Witten, I.H., & Eibe, F. (2000). *Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations*. San Fransisco: Morgan Kaufmann.