Natural Language Processing Methods for Automatic Illustration of Text

Johansson, Richard

2006

Natural Language Processing Methods for Automatic Illustration of Text

Richard Johansson

Licentiate Thesis, 2006

Department of Computer Science
Lund Institute of Technology
Lund University

GSLT

Graduate School of Language Technology

ISSN 1652-4691
Licentiate Thesis 4, 2006
LU-CS-LIC:2006-1

Thesis submitted for partial fulfilment of
the degree of licentiate.

Department of Computer Science
Lund Institute of Technology
Lund University
Box 118
SE-221 00 Lund
Sweden

Email: richard@cs.lth.se
WWW: http://www.cs.lth.se/~richard

Typeset using LaTeX 2ε

Printed at E-huset, Lunds Tekniska Högskola.

Abstract

The thesis describes methods for automatic creation of illustrations of natural-language text. The main focus of the work is to convert texts that describe sequences of events in a physical world into animated images. This is what we call *text-to-scene conversion*.

The first part of the thesis describes Carsim, a system that automatically illustrates traffic accident newspaper reports written in Swedish. This system is the first text-to-scene conversion system for non-invented texts.

The second part of the thesis focuses on methods to generalize the NLP components of Carsim to make the system more easily portable to new domains of application. Specifically, we develop methods to sidestep the scarcity of annotated data, needed for training and testing of NLP methods. We present a method to annotate the Swedish side of a parallel corpus with shallow semantic information in the FrameNet standard. This corpus is then used to train a semantic role labeler for Swedish text.

Tack

Mitt största tack går till min handledare Pierre Nugues som alltid ger entusiastisk och kreativ kritik och som låter mig arbeta på det sätt som passar mig bäst: att bränna mig på fingrarna av egna misstag. Tack också till mina kolleger på institutionen för datavetenskap vid Lunds Tekniska Högskola, där detta arbete har utförts sedan december 2003.

I de projekt som beskrivs i denna rapport har vi haft stor nytta av följande externa program och resurser:

- Ordklassmärkaren GRANSKA från Viggo Kann och andra på KTH i Stockholm,
- Dependensparsern MALTPARSER av Joakim Nivre med doktorander vid Växjö universitet,
- Constraintlösaren JACOP av Krzysztof Kuchciński och Radosław Szymanek på institutionen för datavetenskap på LTH,
- Delar av det svenska ordnätet av Åke Viberg och andra på institutionen för lingvistik vid Lunds Universitet.

Jag har haft nöjet att arbeta med en rad examensarbetare: Per Andersson, David Williams, Kent Thureson, Anders Berglund och Magnus Danielsson.

Karin Brundell-Freij, Åse Svensson och András Várhelyi, forskare i trafiksäkerhet vid institutionen för Teknik och Samhälle vid LTH, har bidragit med förslag på hur trafikolyckor ska presenteras i grafisk form.

Margaret Newman-Nowicka förtjänar beröm för att hon lärde mig att formulera mig någorlunda fokuserat, och för många konstruktiva och detaljerade synpunkter.

Projektet Carsim bekostades under 2003 och 2004 delvis av anslag 2002-02380 från Vinnovas språkteknologiprogram.

Slutligen går mitt tack till mina föräldrar och bror samt Piret, Henn och Svetlana Saar.

Contents

List of Figures

List of Tables

Chapter 1

Introduction: Context and Overview

For many types of text, a proper understanding requires the reader to form some sort of mental images. This is especially the case for texts describing physical processes. For such texts, *illustrations* are unquestionably very helpful to enable the reader to understand them properly. As it has been frequently noted, it is often easier to explain physical phenomena, mathematical theorems, or structures of any kind using a drawing than words, and this can be seen from almost any textbook. The connection between image and understanding has long been noted by cognitive scientists, but has received comparatively little attention in the area of automatic natural language understanding.

While it is clear that illustrations are helpful for understanding a text, the process of creating them by hand may be laborious and costly. This task is usually performed by graphic artists.

The central aim of this thesis is to survey and develop methods for performing this process automatically. Specifically, we focus on what is necessary to perform automatic illustration of texts describing sequences of events in a physical world, which we call *text-to-scene conversion*.

1.1 Related Work

Prototypes of text-to-scene conversion systems have been developed in a few projects. The earliest we are aware of is the CLOWNS system

(Simmons, 1975), a typical example of 1970s microworld research. It processed simple spatial descriptions of clowns and their actions, such as *A clown holding a pole balances on his head in a boat.* The texts were written in a restricted form of English. The system could also produce simple animations of motion verbs in sentences such as *A clown on his head sails a boat from the dock to the lighthouse.* NALIG (Adorni et al., 1984) was another early system. It could handle simple fragments of sentences written in Italian describing spatial relations between objects. Both the CLOWNS system and NALIG applied some elegant qualitative spatial reasoning; however, none of those systems was applicable beyond its microworld.

The most ambitious system to date is the WordsEye project (Coyne and Sproat, 2001), formerly at AT&T and currently at Semantic Light LLC. It produces static 3D images from simple written descriptions of scenes. Its database of 3D models contains several thousands of objects. Unlike all other text-to-scene systems to date, the designers of WordsEye aim at future practical uses of the system: "First and second language instruction; e-postcards; visual chat; story illustrations; game applications; specialized domains, such as cookbook instructions or product assembly." The texts given as examples by the authors are very simple, consisting mostly of a set of spatial descriptions of object placement. However, the system is not intended to handle realistic texts; rather, its purpose is to be a 3D modeling tool with a natural-language user interface.

CogViSys is a system that started with the idea of generating texts from a sequence of video images. The authors found that it could also be useful to reverse the process and generate synthetic video sequences from texts. This text-to-scene converter (Arens et al., 2002) is limited to the visualization of single vehicle maneuvers at an intersection as the one described in this two-sentence narrative: *A car came from Kriegstraße. It turned left at the intersection.* The authors give no further details on what texts they used.

Another ambitious system was the SWAN system (Lu and Zhang, 2002). It converted small fairytales in restricted Chinese into animated cartoons. The system has been used by Chinese television.

Seanchaí/Confucius (Ma and Mc Kevitt, 2003, 2004) is an "intelligent storytelling system". The designers of the system seem to have a high ambition in grounding their representations in semantic and cognitive theory. However, they give very few examples of the kinds of text that the system is able to process, and no description of the results. It appears that the system is able to interpret and animate texts consisting

of a single-event sentence, such as *Nancy gave John a loaf of bread*.

While the previous research has contributed to our understanding of the problem, all the systems suffer from certain shortcomings:

- None of the systems can handle real texts. It seems that the systems are restricted to very simple narratives, typically invented by the designers.
- All systems either produce static images, or treat the temporal dimension of the problem very superficially.
- There has been no indication, let alone any evaluation, of the quality of the systems.

1.2 The Carsim System

The first part of this thesis presents Carsim, a text-to-scene conversion system for traffic accident reports written in Swedish (Johansson et al., 2005, 2004; Dupuy et al., 2001). The traffic accident domain is an example of a genre where the texts are very often accompanied by illustrations. For instance, the US National Transport Safety Board (NTSB) manually produces animated video sequences to illustrate flight, railroad, and road traffic accidents[1]. Additionally, illustrations produced manually are often seen in newspapers, where texts describing road accidents are frequently illustrated using iconic images, at least when the text is long or complex. Thus, we believe that this domain is suitable for a prototype system for automatic illustration.

The Carsim system addresses the shortcomings outlined above:

- It was developed using authentic texts from newspapers.
- It produces animated images, and to do this systematically, the temporal dimension had to be handled carefully.
- The system (the language component and the complete system) was evaluated using real texts.

The architecture of the system is described extensively in Chapter 2. In short, the text is first processed by an Information Extraction (IE) module that produces a symbolic representation. This representation is

[1] See `http://www.ntsb.gov/events/Boardmeeting.htm` for examples.

then converted by a spatio-temporal planner into a complete and unambiguous geometric description that can be rendered (for instance) as 3D graphics.

As a more challenging direction of research, we would like to generalize the system to be able to handle other types of text. However, expecting a system that can handle any kind of text would be naïve. A more realistic goal would be to construct a system that is portable from one restricted domain (such as traffic accident reports) to other restricted domains. The simplifying assumptions that make the process feasible in one domain could then hopefully be adapted or replaced for the new domain. The rest of the thesis deals with the generalization of the language component of Carsim. We do not describe how to generalize the planner, which would probably be at least as complex as for the language component.

1.3 Generalizing the Semantic Components in Carsim

To produce the symbolic representation, Carsim addresses a large number of semantic subproblems. While we were able to reach satisfying results for the traffic accident domain, generalizing the modules that solve those subproblems is non-trivial. They all suffer from being (to varying degrees) domain-specific, i.e. they are either based on *ad hoc* algorithms designed specifically for traffic accidents, or they rely on ontologies or training data that are specific to the domain even though the algorithms are generic.

Because years of failures in large-scale projects have shown us the infeasibility of building generic NLP components by hand (except for a few well-understood areas such as tokenization and morphology), experience tells us that statistical approaches outperform rule-based ones. To construct statistical systems, we need large quantities of data for training. Even if a rule-based system is preferred because a statistical approach is infeasible, resources for evaluation are still necessary.

To construct a semantic resource that can be used for training, we need the following:

- A *theoretical framework*, that is a definition of the structures to describe,
- An *annotation system*, that is a way to encode the relation between a specific text and the structure it represents,

- An *example corpus*, which should preferably be large, unbiased, and have a broad coverage, that demonstrates that the theory and annotation system can be used on real texts, and that can be used to collect the distributional patterns about the particular aspect that we would like to predict automatically.

When using annotated corpora for developing the semantic components in Carsim, the first two criteria were usually satisfied — we based the annotations on existing standards. The third criterion, which is usually very demanding in terms of human effort, was lacking since the texts we used were few and specific to the domain.

In the following subsections, we will describe three central tasks that Carsim addresses, and discuss the resources that could be used to construct more generic components to handle them.

- *Finding predicate arguments*, that is building larger structures from isolated pieces of information.
- *Resolving reference problems*, that is determining the relations between the text and the world it speaks about.
- *Building the temporal structure*, that is finding the temporal relations between the events.

Although these partly intertwined tasks are not sufficient to construct a complete system, we believe that the are crucial for the text-to-scene conversion process in most domains. They are additionally central to applications in other areas as well.

1.3.1 Predicates and Their Arguments

The central task when producing a symbolic representation of the accident is to find the set of predicates that represent the information in the text. In our context, these predicates typically describe the events that constitute the accident. Apart from word sense ambiguity complications (which is usually not a major problem when the domain is restricted) and the reference problems outlined below, finding the predicates that are relevant to the task can be done relatively straightforwardly by checking each word in the text against a domain-specific dictionary that maps words to predicate classes.

A more demanding problem is to link together the isolated pieces of information into larger conceptual structures, that is finding the arguments of each predicate. For example, for each event we must find all

participants involved, when, where, and how it takes place, etc. This is of course crucial for a proper conversion of text into image. Figure 1.1 shows an example of a sentence where the arguments of the predicate *collided* have been marked up.

Tre personer skadades allvarligt när [*två bilar*]$_{\text{ACTORS}}$ ***kolliderade*** [*våldsamt*]$_{\text{MANNER}}$ [*i en korsning*]$_{\text{LOCATION}}$ [*utanför Mariestad*]$_{\text{LOCATION}}$.

Three people were seriously injured when [two cars]$_{\text{ACTORS}}$ [violently]$_{\text{MANNER}}$ **collided** [at an intersection]$_{\text{LOCATION}}$ [near Mariestad]$_{\text{LOCATION}}$.

Figure 1.1: *Predicates and arguments in a sentence.*

Semantic Role Labeling (SRL), that is automatic identification and classification of semantic arguments of predicates, has been a topic of active research for a few years (Gildea and Jurafsky, 2002; Litkowski, 2004; Carreras and Màrquez, 2005, *inter alia)*. The semantic structures based only on predicates and arguments are partial, since they sidestep a number of complicating factors, for example quantification, modality, coreference and similar phenomena such as metonymy, and linking of multiple predicates. However, these "shallow" structures provide a practical and scalable layer of semantics that can be used (and has been used) in real-world NLP applications, for example Information Extraction (Moschitti et al., 2003; Surdeanu et al., 2003) and Question Answering (Narayanan and Harabagiu, 2004). It has also been suggested (Boas, 2002) that it may be of future use in machine translation.

IE systems built using predicate-argument structures are still outperformed by the classical systems based on patterns written by hand (Surdeanu et al., 2003), but the real benefit can be found in development time and adaptability.

Carsim relies on a domain-specific statistical predicate argument classifier. To make a system that is portable to other domains, we need to use a theory and training data that are not specific to the traffic accident domain.

Almost all domain-independent SRL systems have been implemented using the FrameNet or PropBank standards.

FrameNet (Baker et al., 1998) is a large semantic resource, which consists of a lexical database and an example corpus of 130,000 hand-

annotated sentences. It is based on Fillmore's Frame Semantics (1976), which has evolved out of his earlier theory of Case Grammar (1968), the adaptation of the ancient theories of "semantic cases" into the paradigm of Transformational Grammar that was fashionable at the time. While Case Grammar was based on a small set of universal semantic roles such as AGENT, THEME, INSTRUMENT, etc., Frame Semantics uses semantic roles specific to a *frame*. For example, the STATEMENT frame defines the semantic roles SPEAKER, MESSAGE, ADDRESSEE, etc. The frames are arranged in an ontology that defines relations such as inheritance, part-of, and causative-of. Frame Semantics was developed because large-scale annotation showed that the case theory was impractical in many cases. Among other problems, the small set of universal semantic cases proved to be very difficult to define.

The FrameNet designers certainly made an effort to create a well-designed and scalable conceptual model (to this end, FrameNet has been fundamentally redesigned — "reframed" — more than once), but it still remains to be seen how usable FrameNet will be for practical NLP applications[2]. While the annotators have carefully provided lexical evidence for the frames and semantic roles they propose, they only recently started to annotate running text.

An effort that has been partly inspired by the perceived shortcomings of FrameNet is the PropBank project (Palmer et al., 2005), which adds a predicate/argument layer to the Penn Treebank. Unlike FrameNet, the PropBank project is directly aimed at NLP (rather than lexical) research. Its focus has been consistency and complete coverage (by annotating a complete corpus) rather than "interesting examples". Unlike in FrameNet annotation, all arguments and adjuncts of the verbs are annotated as semantic arguments. While PropBank annotates verbal predicates only, the NomBank project (Meyers et al., 2004) annotates the nominal predicates of the Penn Treebank.

1.3.2 Resolving Reference Problems

Several of the semantic components of Carsim address the task of resolving reference problems. This class of problems caused the bulk of the errors made by the Carsim NLP module (Johansson et al., 2004). Carsim handles the following types of reference:

- *Entity identity coreference*: "a car", ..., "the vehicle"
- *Set identity coreference*: "a car and a bus", ..., "the vehicles"

[2]Note, however, that the main purpose of FrameNet is lexical rather than NLP.

- *Subset/element coreference*: "three cars", ..., "two of them", ..., "the first of them"
- *Event coreference*: "a car crashed", ..., "the collision"
- *Metonymy*: "he collided with a tree" (which means that his car hit the tree)
- *Underspecification*: "he was killed in a frontal collision while overtaking" (two vehicles are implicit)

We have developed a robust module (Danielsson, 2005) to solve the first of these problems. This module is generic in design, but relies heavily on a domain-specific ontology and was trained on a set of traffic accident reports. The training data were annotated using the MUC-7 annotation standard (Hirschman, 1997). The problem of identity coreference is intuitively easy to define (two pieces of text refer to "the same thing"), although a careful analysis reveals some potential conceptual fallacies (van Deemter and Kibble, 2000).

The other reference problems are more complex, both to define and to solve. In Carsim, they were addressed by domain-specific *ad hoc* methods. This makes them difficult to port. Generic methods for such problems are rare, although there has been some work on corpus-based and knowledge-based approaches to a few special cases of metonymy (Markert and Hahn, 2002, *inter alia)*.

1.3.3 Ordering Events Temporally

In any text-to-scene conversion system that produces animated sequences of more than one event, the problem of determining the relative positions of the events on the time axis is crucial. This problem may have important applications in other areas as well, such as Question Answering (Moldovan et al., 2005; Saurí et al., 2005).

The complexity of this problem can be realized by considering the following example:

> *Ett par i 40-årsåldern omkom på onsdagseftermiddagen vid en trafikolycka på Öxnehagaleden i Jönköping. Mannen och kvinnan åkte mc och körde in i sidan på en buss som kom ut på leden i en korsning.*
>
> Excerpt from a newswire by TT, August 6, 2003.

> A couple in their forties were killed on Wednesday afternoon in a traffic accident on the Öxnehaga Road in Jönköping. The man and the woman were traveling on a motorcycle and crashed into the side of a bus that entered the lane at a crossroads.
>
> The text above, our translation.

Five events are mentioned in the fragment above. A graph showing the qualitative temporal relations between them in Allen's framework (1984) can be seen in Figure 1.2. As the graph shows, the relation between the order of events in time and their order in discourse may be complex.

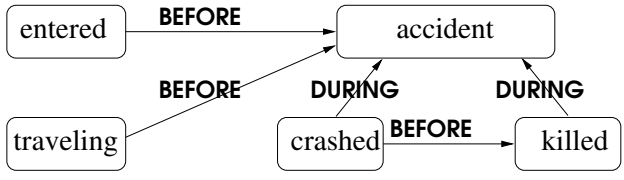

Figure 1.2: *Example of temporal relations in a text.*

To determine the relations between events, Carsim uses a hybrid system, based on one part consisting of hand-written rules and one statistical part based on automatically induced decision trees (Berglund, 2004; Berglund et al., 2006a,b). This system is fairly generic, although it includes a domain-specific list of nominal event expressions. Additionally, it was trained on a relatively small domain-specific set of texts. As for other components of Carsim, the framework needs to be somewhat generalized, and large domain-independent annotated corpora are required for training and evaluation.

The phletora of theories about time and event concepts reflects the complexity of the problem. As shown by Bennett and Galton (2004), it has been approached from many different directions. As can be expected, annotation practices have differed as widely as the theories (Setzer and Gaizauskas, 2001).

TimeML (Pustejovsky et al., 2003a, 2005) is an attempt to create a unified annotation standard for temporal information in text. It is still an evolving standard (the latest annotation guidelines are from October 2005), and TimeBank (Pustejovsky et al., 2003b), the annotated corpus,

is still rather small. Annotation is difficult for humans as well as for machines: human inter-annotator agreement is low, and automatic methods are appearing (Verhagen et al., 2005; Boguraev and Ando, 2005; Mani and Schiffman, 2005, *inter alia)* but have yet to mature. The complex theory, and the fact that a large part of the information to annotate is implicit, accounts for this phenomenon. Whether it is possible to have a good performance in a domain-independent automatic system, and what features will be useful for statistical models, remains to be seen.

Additionally, the question of how to evaluate the performance is still not settled. When evaluating the temporal links produced in Carsim, we used the the method proposed by Setzer and Gaizauskas (2001), which measures precision/recall on the set of links in graphs that have been normalized using the transitive closure of the links.

1.4 Overcoming the Resource Bottlenecks

As the previous section discussed, to construct domain-independent components for semantic processing of text, we need to have broad-coverage annotated resources that can be used for model estimation and validation. For other languages than English and possibly a few others, these resources are very rarely available. Since annotation by hand is expensive in terms of human effort and time, we would like to develop methods that at least partly allow us to sidestep this resource scarcity.

One option that has been used in a number of projects is to use unsupervised learning methods. For instance, Swier and Stevenson (2004, 2005) describe an experiment in an unsupervised method to train a SRL system from unannotated text. When completely unsupervised methods are infeasible, bootstrapping methods such as Yarowsky's algorithm (1995) can instead be used. In those methods, a small hand-annotated training set is used to train a system that annotates a larger set, out of which the method selects the data of highest quality to extend the training set.

Another alternative that has received attention for a few years is to produce annotated training data in a new language by automatic means by making use of manually annotated training data produced in English.

In Chapters 3 and 4, we describe a method to construct a generic FrameNet-based SRL system that could be used to replace the predi-

cate/argument module in Carsim. To produce training data, we applied methods for automatic projection of FrameNet data across languages. In those experiments, an English FrameNet-based SRL system was trained on the FrameNet example corpus, and applied on the English side of the Europarl parallel corpus (Koehn, 2005). By using a word aligner, the annotation could then be transferred to the other side of the parallel corpus. In Chapter 3, we describe a tentative study of the quality of the transferred data, and in Chapter 4, we use such data to train a Swedish SRL system. This system was finally evaluated on a manually translated portion of the FrameNet examle corpus.

We used FrameNet for the experiment rather than PropBank, despite the healthier annotation practices of PropBank, since we believe that the FrameNet concepts make sense across languages. The frames in FrameNet are motivated by cognitive theory, while PropBank is defined closely to English syntax — it is to a certain extent based on Levin's work on verb classes in English (1993). In addition, PropBank is only defined for verbs. As described in Chapter 4, we made three assumptions that are necessary for an automatic transfer of FrameNet annotation to be meaningful:

- The English FrameNet (i.e. the set of frames, their sets of semantic roles, and the relations between the frames) is meaningful in the target language (in our case Swedish) as well.

- When a word belonging to a certain frame is observed on the English side of the parallel corpus, it has a counterpart on the target side belonging to the same frame.

- Some of the predicate arguments on the English side have counterparts with the same semantic roles on the target side.

These assumptions may all be put into question. In particular, the second assumption will fail in many cases because the translations are not literal, which means that the same information may be expressed very differently in the two languages. In addition, there may be no exact counterpart of an English word in Swedish. These complications, in addition to mistakes made in projection and when applying the English SRL system, introduce noise into the training data. The aim of the experiment is to determine how well such noisy training data can be used to train a SRL system.

1.5 Overview of the Thesis

Chapter 2 deals with a text-to-scene conversion system for the traffic accident domain. The chapter mainly consists of material from three articles (Johansson et al., 2004, 2005; Johansson and Nugues, 2005a).

Chapter 3 is based on an article in the Romance FrameNet workshop (Johansson and Nugues, 2005b) and describes how FrameNet annotation can be automatically transferred across languages using parallel corpora.

Chapter 4, based on a forthcoming article (Johansson and Nugues, 2006b), describes how such data are used to construct of a FrameNet argument identifier and classifier for Swedish text.

Chapter 5 concludes the thesis and describes some possible directions of future research.

Chapter 2

Text-to-Scene Conversion in the Traffic Accident Domain

In this chapter, we describe a system that automatically converts narratives into 3D scenes. The texts, written in Swedish, describe road accidents. One of the key features of the program is that it animates the generated scene using temporal relations between the events. We believe that this system is the first text-to-scene converter that is not restricted to invented narratives.

The system consists of three modules: natural language interpretation based on IE methods, a planning module that produces a geometric description of the accident, and finally a visualization module that renders the geometric description as animated graphics.

An evaluation of the system was carried out in two steps: First, we used standard IE scoring methods to evaluate the language interpretation. The results are on the same level as for similar systems tested previously. Secondly, we performed a small user study to evaluate the quality of the visualization. The results validate our choice of methods, and since this is the first evaluation of a text-to-scene conversion system, they also provide a baseline for further studies.

The structure of this chapter is as follows: Section 2.1 describes the system and the application domain. Section 2.2 details the implementation of the natural language interpretation module. Then, in Section 2.3, we turn to the spatial and temporal reasoning that is needed to con-

struct the geometry of the scene. The evaluation is described in Section 2.4. Finally, we discuss the results and their implications in Section 2.5.

2.1 The Carsim System

Narratives of a car accidents often make use of descriptions of spatial configurations, movements, and directions that are sometimes difficult to grasp for readers. We believe that forming consistent mental images is necessary to understand such texts properly. However, some people have difficulties in imagining complex situations and may need visual aids pre-designed by professional analysts.

Carsim is a computer program[1] that addresses this need. It is intended to be a helpful tool that can enable people to imagine a traffic situation and understand the course of events properly. The program analyzes texts describing car accidents and visualizes them in a 3D environment.

To generate a 3D scene, Carsim combines natural language processing components and a visualizer. The language processing module adopts an IE strategy and includes machine learning methods to solve coreference, classify predicate/argument structures, and order events temporally. However, as real texts suffer from underspecification and rarely contain a detailed geometric description of the actions, information extraction alone is insufficient to convert narratives into images automatically. To handle this, Carsim infers implicit information about the environment and the involved entities from key phrases in the text, knowledge about typical traffic situations, and properties of the involved entities. The program then uses a visualization planner that applies spatial and temporal reasoning to find the simplest configuration that fits the description of the entities and actions described in the text and to infer details that are obvious when considering the geometry but unmentioned in the text.

2.1.1 A Corpus of Traffic Accident Descriptions

Carsim has been developed using authentic texts. As a development set, we collected approximately 200 reports of road accidents from various Swedish newspapers. The task of analyzing the news reports is made more complex by their variability in style and length. The size of

[1] An online demonstration of the system is available as a Java Webstart application at `http://www.lucas.lth.se/lt`.

the texts ranges from a couple of sentences to more than a page. The amount of details is overwhelming in some reports, while in others, most of the information is implicit. The complexity of the accidents ranges from simple crashes with only one car to multiple collisions with several participating vehicles and complex movements. Although our work has focused on the press clippings, we also have access to accident reports, written by victims, from the STRADA database (Swedish TRaffic Accident Data Acquisition, see Karlberg, 2003) of Vägverket, the Swedish traffic authority.

The next text is an excerpt from our test corpus. The report is an example of a press wire describing an accident.

> ***Olofströmsyngling omkom i trafikolycka***
> *En 19-årig man från Olofström omkom i en trafikolycka mellan Ingelstad och Väckelsång. Singelolyckan inträffade då mannen, i mycket hög hastighet, gjorde en omkörning.*
>
> *Det var på torsdagseftermiddagen förra veckan som den 19-årige olofströmaren var på väg mot Växjö. På riksväg 30 mellan Ingelstad och Väckelsång, två mil söder om Växjö, gjorde mannen en omkörning i mycket hög hastighet. När 19-åringen kom tillbaka på rätt sida av vägen efter omkörningen fick han sladd på bilen, for över vägen och ner i ett dike där han i hög fart kolliderade med en sten. Bilen voltade i samband med kollisionen och började brinna. En förbipasserande bilist stannade och släckte elden, men enligt Växjöpolisen visade mannen inga livstecken efter olyckan. Med all sannolikhet omkom 19-åringen omedelbart i och med den kraftiga kollisionen.*
>
> Blekinge Läns Tidning, October 18, 2004.

> **Youth from Olofström Killed in Traffic Accident**
> A 19-year-old man from Olofström was killed in a traffic accident between Ingelstad and Väckelsång. The single accident occured when the man overtook at a very high speed.
>
> The incident took place in the afternoon last Thursday, when the youth was traveling towards Växjö. On Road 30 between Ingelstad and Väckelsång, 20 kilometers south of Växjö, he overtook at a high velocity. While steering back to the right side of the road, the vehicle skidded across the road into a ditch where it collided with a rock. The car overturned and caught fire. A traveler passing by stopped and put out the fire, but according to the Växjö Police, the man showed no signs of life after the accident. In all probability, he was killed immediately due to the violent collision.
>
> The text above, our translation.

2.1.2 Architecture of the Carsim System

We use a division into modules where each one addresses one step of the conversion process (see Figure 2.1).

- A *natural language processing* module that interprets the text to produce an intermediate symbolic representation.
- A *spatio-temporal planning and inference* module that produces a full geometric description given the symbolic representation.
- A *graphical* module that renders the geometric description as graphics.

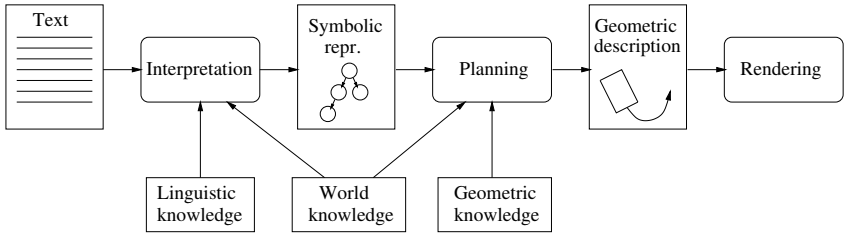

Figure 2.1: *System architecture of Carsim.*

We use the intermediate representation as a bridge between texts and geometry. This is made necessary because the information expressed by most reports usually has little affinity with a geometric description. Exact and explicit accounts of the world and its physical properties are rarely present. In addition, our vocabulary is finite and discrete, while the set of geometric descriptions is infinite and continuous.

Once the NLP module has interpreted and converted a text, the planner maps the resulting symbolic representation of the world, the entities, and behaviors, onto a complete and unambiguous geometric description in a Euclidean space.

Certain facts are never explicitly stated, but are assumed by the author to be known to the reader. This includes linguistic knowledge, world knowledge (such as traffic regulations and typical behaviors), and geometric knowledge (such as typical sizes of vehicles). The language processing and planning modules take this knowledge into account in order to produce a credible geometric description that can be visualized by the renderer.

2.1.3 The Symbolic Representation

The symbolic representation has to manage the following trade-off. In order to be able to describe a scene, it must contain enough information to make it feasible to produce a consistent geometric description that is acceptable to the user. On the other hand, to be able to capture the relevant information in the texts, the representation has to be close to ways human beings describe things.

The representation is implemented using Minsky-style ("object-oriented") frames, which means that the objects in the representation consist of a type and a number of predefined attribute/value slots. The ontologies defining the types were designed with assistance of traffic safety experts. The representation consists of the following four concept categories:

- *Objects*. These are typically the physical entities that are mentioned in the text, but we might also need to present abstract or oneiric entities as symbols in the scene. Each object has a type that is selected from a predefined, finite set. `Car` and `Truck` are examples of object types.

- *Events*, in this context corresponding to the possible object behaviors, are also represented as entities with a type from a predefined set. `Overturn` and `Impact` are examples. The concept of an event, by which we intuitively mean an activity that goes on during a point or period in time, is difficult to define formally (see Bennett and Galton (2004) for a recent discussion).

- *Spatial and Temporal Relations*. The objects and the events need to be described and related to each other. The most obvious examples of such information are *spatial* information about objects and *temporal* information about events. We should be able to express not only exact quantities, but also qualitative information (by which we mean that only certain fundamental distinctions are made). Suitable systems of expressing these concepts are the positional and topological systems described by Cohn and Hazarika (2001) and Allen's temporal relations (Allen, 1984). `Behind`, `FromLeft`, and `During` are examples of spatial and temporal relations used in Carsim.

- *Environment*. The environment of the accident is important for the visualization to be understandable. Significant environmental parameters include light, weather, road conditions, and type

of environment (such as rural or urban). Another important parameter is topography, but we have set it aside since we have no convenient way to express this qualitatively.

Although we have made no psychological experiments on the topic, we think that the representation is easy to understand and relatively close to how humans perceive the world. All concepts used in the symbolic representation are grounded, i.e. explicitly defined in terms of how they should be realized as graphics. However, we are aware that the representation contains some ontological flaws. For example, as noted by Heraclitus, the identity of an "object" is problematic since objects may split or merge over time, or evolve into something completely different (see also Bennett, 2002). Equally problematic is the somewhat rigid notion of "types" of objects and events. In addition, modal expressions such as *she tried* or *she was forced* are not represented (however, it is difficult to imagine how such constructions would be graphically represented). It should, however, be noted that we do not intend to come up with a theory about the nature of the world, but rather a practical way to process statements made by humans.

2.2 Natural Language Interpretation

We use IE techniques to interpret the text. This is justified by the symbolic representation, which is restricted to a limited set of types and the fact that only a part of the meaning of the text needs to be presented visually. The IE module consists of a sequence of components (Figure 2.2). The first components carry out a shallow parse: POS tagging, NP chunking, complex word recognition, and clause segmentation. This is followed by a cascade of semantic markup components: named entity recognition, detection and interpretation of temporal expressions, object markup and coreference, and predicate argument detection. Finally, the marked-up structures are interpreted, i.e. converted into a symbolic representation of the accident. The development of the IE module has been made more complex by the fact that few tools or annotated corpora are available for Swedish. The only significant external tool we have used is the Granska part-of-speech tagger (Carlberger and Kann, 1999).

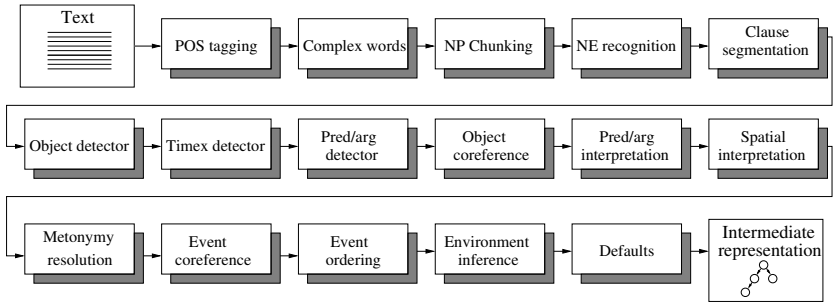

Figure 2.2: *Architecture of the language interpretation module.*

2.2.1 Entity Detection and Coreference

A correct detection of the entities involved in the accident is crucial for the graphical presentation to be understandable. We first locate the likely participants among the noun phrases in the text by checking their heads against a dictionary that maps words to concepts in the ontology. The dictionary was partly constructed using a fragment of the Swedish WordNet (Viberg et al., 2002). We then apply a coreference solver to link the groups that refer to identical entities. This results in a set of equivalence classes referring to entities that are likely to be participants in the accident.

The coreference solver uses a hand-written filter in conjunction with a statistical system based on decision trees (Danielsson, 2005). The filter first tests each antecedent-anaphor pair using 12 grammatical and semantic features to prevent unlikely coreference. The statistical system, which is based on the model described by Soon et al. (2001), then uses 20 features to classify pairs of noun groups as coreferring or not. These features are lexical, grammatical, and semantic. The trees were induced from a set of hand-annotated examples using the ID3 algorithm.We implemented a novel *feature transfer* mechanism that propagates and continuously changes the values of semantic features in the coreference chains during clustering. This means that the coreferring entities inherit semantic properties from each other. Feature transfer, as well as domain-specific semantic features, proved to have a significant impact on the performance.

2.2.2 Domain Events

In order to produce a symbolic representation of the accident, we need to recreate the course of events. We find the events using a two-step procedure. First, we identify and mark up text fragments that describe events, and locate and classify their arguments. Secondly, the fragments are interpreted, i.e. mapped into the symbolic representation, to produce event objects as well as the involved participants, spatial and temporal relations, and information about the environment. This two-step procedure is similar to other work that uses predicate-argument structures for IE (see for example Surdeanu et al., 2003).

We classify the arguments of each predicate (assign them a semantic role) using a statistical system, which was trained on about 900 hand-annotated examples. Following Gildea and Jurafsky (2002), there has been a relative consensus on the features that the classifier should use. However, we did not use a full parser and we avoided features referring to the parse tree. Also, since the system is domain-specific, we have introduced an ontology-dependent *semantic type* feature that takes the following values: dynamic object, static object, human, place, time, cause, or speed.

Similarly to the method described by Gildea and Jurafsky (2002), the classifier chooses the role that maximizes the estimated probability of a role given the values of the target, head, and semantic type attributes:

$$\hat{P}(r|t, head, sem) = \frac{C(r, t, head, sem)}{C(t, head, sem)}.$$

If a particular combination of target, head, and semantic type is not found in the training set, the classifier uses a back-off strategy, taking the other attributes into account. In addition, we tried other classification methods (ID3 with gain ratio and Support Vector Machine (SVM)) without any significant improvement.

When the system has located the references to domain events in the text, it can interpret them; that is, we map the text fragments to entities in the symbolic representation. This stage makes significant use of world knowledge, for example to handle relationships such as metonymy and ownership.

Since it is common that events are mentioned more than once in the text, we need to remove the duplicates when they occur. Event coreference is a task that can be treated using similar methods as those we used for object coreference. However, event coreference is a simpler problem since the range of candidates is narrowed by the involved participants

and the event type. To get a minimal description of the course of events, we have found that it is sufficient to unify (i.e. merge) as many events as possible, taking event types and participants into account. To complete the description of the events and participants, we finally apply a set of simple default rules and heuristics to capture the information that is lacking due to mistakes or underspecification.

2.2.3 Temporal Ordering of Events

Since we produce an animation rather than just a static image, we have to take time into account by determining the temporal relations between the actions that are described in the text. Although the planner alone can infer a probable course of events given the positions of the participants, and some orderings are deducible by means of simple *ad hoc* rules that place effects after causes (such as a fire after a collision), we have opted for a general approach.

We developed a component (see Berglund (2004); Berglund et al. (2006a,b) for implementation details) based on TimeML (Pustejovsky et al., 2003a, 2005), which is a generic framework for markup of temporal information in text. We first create an ordering of all events in the text (where all verbs, and a small set of nouns, are considered to refer to events) by generating temporal links (orderings) between the events. The links are generated by a hybrid system consisting of a statistical system based on decision trees and a small set of hand-written heuristics.

The statistical system considers events that are close to each other in the text, and that are not separated by explicit temporal expressions. It was trained on a set of hand-annotated examples consisting of 476 events and 1,162 temporal relations. The decision trees were produced using the C4.5 tool (Quinlan, 1993) and make use of the following information:

- *Tense*, *aspect*, and *grammatical construct* of the verb groups that denote the events.
- *Temporal signals* between the words. This is a TimeML term for temporal connectives and prepositions such as *when*, *after*, and *during*.
- *Distance* between the words, measured in tokens, sentences, and in punctuation signs.

The range of possible output values is the following subset of the temporal relations proposed by Allen (1984): *simultaneous*, *after*, *before*, *is_included*, *includes*, and *unspecified*.

After the decision trees have been applied, we remove conflicting temporal links using probability estimates derived from C4.5. We use a greedy loop removal strategy that adds links in an order determined by the probabilities of the links, and ignores the links that introduce conflicts into the graph due to violated transitivity relations. As a final step, we extract the temporal relations between the events that are relevant to Carsim.

2.2.4 Inferring the Environment

The environment is important for the graphical presentation to be credible. We use traditional IE techniques, such as domain-relevant patterns, to find explicit descriptions of the environment.

Additionally, as noted by the WordsEye team (Sproat, 2001), the environment of a scene may often be obvious to a reader even though it is not explicitly referred to in the text. In order to capture this information, we try to infer it by using prepositional phrases that occur in the context of the events, which are used as features for a classifier. We then use a Naïve Bayesian classifier to guess whether the environment is urban or rural.

2.3 Planning the Animation

We use a planning system to create the animation out of the extracted information. It first determines a set of constraints that the animation needs to fulfil. Then, it goes on to find the initial directions and positions. Finally, it uses a search algorithm to find the trajectory layout. Since we use no backtracking between the processing steps in the planning procedure, the separation into steps introduces a risk. However, it reduces the computation load and proved sufficient for the texts we considered, enabling an interactive generation of 3D scenes.

2.3.1 Finding the Constraints

The constraints on the animation are defined using the detected events and the spatial and temporal relations combined with the implicit and domain-dependent knowledge about the world. The events are expressed as conjunctions of primitive predicates about the objects and

their behavior in time. For example, if there is an `Overtake` event where O_1 overtakes O_2, this is translated into the following proposition:

$$\exists t_1, t_2. MovesSideways(O_1, Left, t_1) \wedge Passes(O_1, O_2, t_2) \wedge t_1 < t_2$$

In addition, other constraints are implied by the events and our knowledge of the world. For example, if O_1 overtakes O_2, we add the constraints that O_1 is initially positioned behind O_2, and that O_1 has the same initial direction as O_2. Other constraints are added due to the non-presence of events, such as

$$NoCollide(O_1, O_2) \equiv \neg \exists t. Collides(O_1, O_2, t)$$

if there is no mentioned collision between O_1 and O_2. Since we assume that all collisions are explicitly described in the texts, we don't want the planner to add more collisions even if that would make the trajectories simpler.

2.3.2 Finding Initial Directions and Positions

We use constraint propagation techniques to infer initial directions and positions for all the involved objects. We first set those directions and positions that are stated explicitly. Each time a direction is uniquely determined, it is set and this change propagates to the sets of available choices of directions for other objects whose directions have been stated in relation to the first one. When the direction cannot be determined uniquely for any object, we pick one object and set its direction. This goes on until the initial directions have been inferred for all objects. A similar procedure is applied to determine the initial positions of the vehicles.

2.3.3 Finding the Trajectories

After the constraints have been found, we use the IDA* search method (Korf, 1985) to find a optimal trajectory layout, that is a layout that is as simple as possible while violating no constraints. The IDA* method is an iterative deepening best-first search algorithm that uses a heuristic function to guide the search. As heuristic, we use the number of violated constraints multiplied by a constant in order to keep the heuristic admissible (i.e. not overestimate the number of modifications to the trajectory that are necessary to reach a solution).

The most complicated traffic accident report in our development corpus contains 8 events, which results in 15 constraints during search, and needs 6 modifications of the trajectories to arrive at a trajectory layout that violates no constraints. This solution is found in a few seconds. Most accidents can be described using only a few constraints.

At times, no solution is found within reasonable time. This may, for instance, happen when the IE module has produced incorrect results. In this case, the planner backs off. It first relaxes some of the temporal constraints (for example: $Simultaneous$ constraints are replaced by $NearTime$). Next, all temporal constraints are removed.

2.4 Evaluation

We evaluated the components of the system, first by measuring the quality of the extracted information using standard IE evaluation methods, then by performing a user study to determine the overall perception of the complete system. For both evaluations, we used 50 previously unseen texts, which had been collected from newspaper sources on the web. The size of the texts ranged from 36 to 541 tokens.

2.4.1 Evaluation of the Information Extraction Module

For the IE module, three aspects were evaluated: object detection, event detection, and temporal relations between correctly detected events. Table 2.1 shows the precision and recall figures.

	P	R	$F_{\beta=1}$
Objects	0.96	0.86	0.91
Events	0.86	0.85	0.85
Temporal relations	0.73	0.55	0.62

Table 2.1: *Statistics for the IE module on the test set.*

The evaluations of object and event detection were rather straightforward. A road object was considered to be correctly detected if a corresponding object was either mentioned in or implicit from the text, and the type of the object was correct. The same criteria applied to the detection of events, but here we also added the criterion that the actor (and victim, where it applies) must be correct.

Evaluating the quality of the temporal orderings proved to be less straightforward. First, to make it feasible to compare the graph of orderings to the correct graph, it must be converted to some normal form. For this, we used the transitive closure (that is, we made all implicit links explicit). The transitive closure has some drawbacks; for example, one small mistake may cause a large impact on the precision and recall measures if many events are involved. However, we found no other obvious method for normalizing the temporal graphs.

A second problem when evaluating temporal orderings is how to handle links between incorrectly detected events. For example, this is the case when event coreference resolution fails and multiple instances of the same event are reported. In this study, we only count links between correctly detected events.

The results of the event detection are comparable to those reported in previously published work. Surdeanu et al. (2003) report an F-measure of 0.83 in the Market Change domain for a system that uses similar IE methods.[2] Although our system has a different domain, a different language, and different resources (their system is based on PropBank), the figures are roughly similar. The somewhat easier task of detecting the objects results in higher figures, demonstrating that the method chosen works satisfactorily for the task at hand.

For the crucial task of determining the temporal relations between the events, the figures leave room for improvement. Still, the figures for this complex task are significantly better than for a trivial system that assumes that the temporal order is identical to the narrative order (see Berglund et al. (2006a,b) for a more comprehensive discussion). It should also be added that as far as we are aware, this is the first practical application of automatic detection of temporal relations in an IE system.

2.4.2 User Study to Evaluate the Visualization

Four users were shown the animations of subsets of the 50 test texts. Figure 2.3 shows an example corresponding to the text from Subsection 2.1.1. The users graded the quality of animations using the following scores: 0 for wrong, 1 for "more or less" correct, and 2 for perfect. The average score was 0.91. The number of texts that had an average score of 2 was 14 (28 percent), and the number of texts with an average score of at least 1 was 28 (56 percent). While the figures are far from perfect for a fully automatic system, they may be more than acceptable for

[2]We have assumed that the Templettes that they use roughly can be identified with events.

a semi-automatic system that may reduce development time in media production — the typical result of the system is a "more or less" correct result that may be post-processed by a user. Since this is the first quantitative study of the impression of a text-to-scene conversion system, the figures provide a baseline that may be of use in further studies. However, comparisons of systems in different domains, where the degree of pragmatic complexity may vary considerably, must of course be taken with a grain of salt.

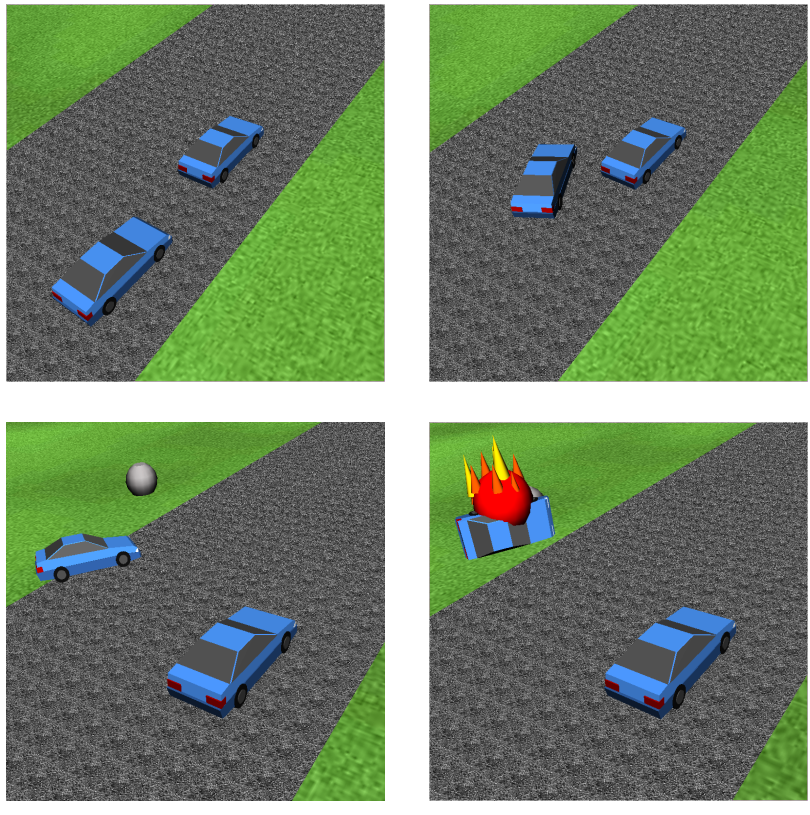

Figure 2.3: *Screenshots from the animation of the example text.*

We calculated the pairwise inter-rater agreement using the weighted κ coefficient (Cohen, 1960) with quadratic weights, for which we obtained the value 0.73 (as a rule of thumb, a value above 0.70 is usually considered a good agreement). Additionally, we calculated the per-text

standard deviation (SD)[3] and obtained the value of 0.45, which is significantly lower[4] than the SD for a randomized sample from the same distribution (0.83). Finally, we calculated the pairwise correlation of the annotations and obtained the value 0.75. All measures suggest that the agreement among annotators is enough for the figures to be relevant.

During discussions with users, we had a number of unexpected opinions about the visualizations. One important example of this is what kind of implicit information they infer from reading the texts. For example, given a short description of a crash in an urban environment, one user imagined a collision of two moving vehicles at an intersection, while another user interpreted it as a collision between a moving and a parked car.

This user response shows that the task of imagining a situation is difficult for humans as well as for machines. Furthermore, while some users have suggested that we improve the realism (for example, the physical behavior of the objects), discussions generally made it clear that the semi-realistic graphics that we use (see Figure 2.3) may suggest to the user that the system knows more than it actually does. Since the system visualizes symbolic information, it may actually be more appropriate to present the graphics in a more "abstract" manner that reflects this better, for example via symbolic signs in the scene. How the information should be presented visually to the user in order to assist understanding as well as possible is a deep cognitive problem that we cannot answer.

2.5 Conclusion and Perspectives

We have presented a system based on information extraction and symbolic visualization that enables to convert real texts into 3D scenes. As far as we know, Carsim is the only text-to-scene conversion system that has been developed and tested using non-invented narratives. It is also unique in the sense that it produces animated graphics by taking temporal relations between events into account.

We have provided the first quantitative evaluation of a text-to-scene conversion system, which shows promising results that validate our

[3] We calculated this using the formula $SD = \sqrt{\frac{\sum (x_{ij} - \dot{x}_i)^2}{\sum (n_i - 1)}}$, where x_{ij} is the score assigned by annotator j on text i, $\dot{x}_i$ the average score on text i, and n_i the number of annotators on text i.

[4] An approximate upper bound at the 95% level is $SD \cdot \sqrt{f/\chi^2_{0.95}(f)} = 0.58$, where $f = \sum (n_i - 1) = 29$.

choice of methods and set a baseline for future improvements. Although the figures are somewhat low for a fully automatic system, we believe that they are perfectly acceptable in a semi-automatic context.

As a possible future project, we would like to extend the prototype system to deeper levels of semantic information. While the current prototype uses no external knowledge, we would like to investigate whether it is possible to integrate additional knowledge sources in order to make the visualization more realistic and understandable. Two important examples of this are geographical and meteorological information, which could be helpful in improving the realism and in creating a more accurate reconstruction of the circumstances and the environment. Another topic that has been prominent in our discussions with traffic safety experts is how to reconcile different narratives that describe the same accident.

We believe that although it is certainly impossible to create a truly general system, the architecture and IE-based strategy makes it feasible to construct systems that are reasonably portable across domains and languages. The limits are set by the complexity of the domain and the availability of knowledge resources, such as databases of object geometries, ontologies, and annotated corpora.

Chapter 3

Cross-language Transfer of FrameNet Annotation

We present a method for producing FrameNet annotation for new languages automatically. The method uses sentence-aligned corpora and transfers bracketing of target words and frame elements using a word aligner.

The system was tested on an English-Spanish parallel corpus. On the task of projection of target word annotation, the system had a precision of 69% and a recall of 59%. For sentences with non-empty targets, it had a precision of 84% and a recall of 81% on the task of transferring frame element annotation. The approximate precision of the complete frame element bracketing of Spanish text was 64%.

3.1 Introduction

The availability of annotated corpora such as FrameNet (Baker et al., 1998), PropBank (Palmer et al., 2005), MUC (Hirschman, 1997), and TimeBank (Pustejovsky et al., 2003b), has played an immense role in the recent development of automatic systems for semantic processing of text. While manually annotated corpora of high quality exist for English, this is a scarce resource for smaller languages. Since the size of the training corpus is of utmost importance, this could significantly impair the quality of the corresponding language processing tools. All things being equal, the corpus size is the key factor to improve accuracy (Banko and Brill, 2001). Given the annotation cost, it is unrealistic

to believe that hand-annotated corpora in smaller languages will ever reach the size of their equivalent counterparts in English.

This article describes an automatic system for FrameNet annotation (target words and frame elements) of texts in new languages. It uses an English SRL system to automatically annotate the English sentences in a parallel corpus. A word aligner is then used to transfer the marked-up entities to the target language. We describe results of the system applied to an English-Spanish parallel corpus taken from the proceedings of the European Parliament (Koehn, 2005).

3.2 Background to FrameNet

Frame semantics (Fillmore, 1976) is a framework that focuses on the relations between lexical meanings — lexical units — and larger conceptual structures — *semantic frames*, typically referring to situations, states, properties or objects. It comes as a development of Fillmore's earlier theory of semantic cases.

FrameNet (Baker et al., 1998) is a comprehensive lexical database that lists frame-semantic descriptions of English words. It consists of a set of frames, which are arranged in an ontology using relations such as inheritance, part-of, and causative-of. Different senses of ambiguous words are represented by different frames. For each frame, FrameNet lists a set of lemmas (nouns, verbs, and adjectives). When such a word occurs in a sentence, it is called a *target word* that *evokes* the frame.

Properties of and participants in a situation are described using *frame elements*, each of which has a *semantic role* from a small frame-specific set, which defines the relation of the Frame Element (FE) to the target word.

In addition, FrameNet comes with a large set of manually annotated example sentences, which are typically used by statistical systems for training and testing. Figure 3.1 shows an example of such a sentence. In that example, the word *statements* has been annotated as a target word evoking the STATEMENT frame, as well as two FEs relating to that target word (SPEAKER and TOPIC).

As usual in these cases, [both parties]$_{\text{SPEAKER}}$ agreed to make no further **statements** [on the matter]$_{\text{TOPIC}}$.

Figure 3.1: *A sentence from the FrameNet example corpus.*

3.3 Related Work

Parallel corpora are now available for many language pairs. Annotated corpora are much larger in English, which means that language processing tools, including parsers, are generally performing better for this language. Hwa et al. (2002) applied a parser on the English part of a parallel corpus and projected the syntactic structures on texts in the second language. They reported results that rival commercial parsers. Diab and Resnik (2002) also used them to disambiguate word senses.

Yarowsky et al. (2001) describe a method for cross-language projection, using parallel corpora and a word aligner, that is applied to a range of linguistic phenomena, such as named entities and noun chunk bracketing. This technique is also used by Riloff et al. (2002) to transfer annotations for IE systems.

Recently, these methods have been applied to FrameNet annotation. Padó and Lapata (2005a) use projection methods, and a set of filtering heuristics, to induce a dictionary of FrameNet target words (frame evoking elements). In a later article (Padó and Lapata, 2005b), they give a careful and detailed study of methods of transferring semantic role information. However, they crucially rely on an existing FrameNet for the target language (in their case German) to select suitable sentence pairs, and the source-language annotation was produced by human annotators.

A rather different method to construct a bilingual FrameNet is the approach taken by BiFrameNet (Chen and Fung, 2004; Fung and Chen, 2004). In that work, annotated structures in a new language (in that case Chinese) are produced by mining for similar structures rather than projecting them via parallel corpora.

3.4 Automatic Transfer of Annotation

3.4.1 Motivation

Although the meaning of the two sentences in a sentence pair in a parallel corpus should be roughly the same, a fundamental question is whether it is meaningful to project semantic markup of text across languages. Equivalent words in two different languages sometimes exhibit subtle but significant semantic differences. However, we believe that a transfer makes sense, since the nature of FrameNet is rather coarse-grained. Even though the words that evoke a frame may not have exact counterparts, it is probable that the frame itself has.

For the projection method to be meaningful, we must make the following assumptions:

- The complete frame ontology in the English FrameNet is meaningful in Spanish as well, and each frame has the same set of semantic roles and the same relations to other frames.
- When a target word evokes a certain frame in English, it has a counterpart in Spanish that evokes the same frame.
- Some of the FEs on the English side have counterparts with the same semantic roles on the Spanish side.

In addition, we made the (obviously simplistic) assumption that the contiguous entities we project are also contiguous on the target side.

These assumptions may all be put into question. Above all, the second assumption will fail in many cases because the translations are not literal, which means that the sentences in the pair may express slightly different information. The third assumption may be invalid if the information expressed is realized by radically different constructions, which means that an argument may belong to another predicate or change its semantic role on the Spanish side. Padó and Lapata (2005b) avoid this problem by using heuristics based on a target-language FrameNet to select sentences that are close in meaning. Since we have no such resource to rely on, we are forced to accept that this problem introduces a certain amount of noise into the automatically annotated corpus.

3.4.2 Producing and Transferring the Bracketing

Using well-known techniques (Gildea and Jurafsky, 2002; Litkowski, 2004), we trained an SVM-based SRL system using 25000 randomly selected sentences from FrameNet. On a test set from FrameNet, we estimated that our labeler has a precision of 0.72 and a recall of 0.63. The result is slightly lower than the systems at Senseval (Litkowski, 2004), possibly because we used all frames from FrameNet rather than a subset, and that we did not assume that the frame was known a priori.

We used the Europarl corpus (Koehn, 2005) and the included sentence aligner, which uses the Gale-Church algorithm. We removed the instances where one English sentence was mapped to more than one in the target language, and for each pair of sentences, a word alignment was produced using GIZA++ (Och and Ney, 2003). Figure 3.2 shows an example of a sentence pair with word alignment. Since we are transferring bracketing from English, the word aligner maps each token to

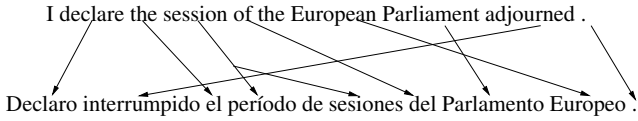

Figure 3.2: *Word alignment example.*

a set of Spanish tokens, but not the other way round. This is why in the figure, only the second English token, rather than the first two, is mapped onto the first Spanish.

For the experiment described here, we labeled 50 English sentences and transferred the annotation to the Spanish sentences. We first located the target words on the English side. Since we did not have access to a reliable word sense disambiguation module, we used all words that occurs as a target in the FrameNet example corpus at least once. Secondly, FEs were produced for each target. 240 target words were found on the English side. Since we did not assume any knowledge of the frame, we used the available semantic roles for all possible interpretations of the target word as features for the classifier. We ignored some common auxiliary verbs: *be*, *have*, *do*, and *get*.

For each entity (target word or FE), we found the target-language counterpart using the maximal span of all the words within the bracketing. We added the constraint that FEs should not cross the target word (in that case, we just used the part that was to the right of the target).

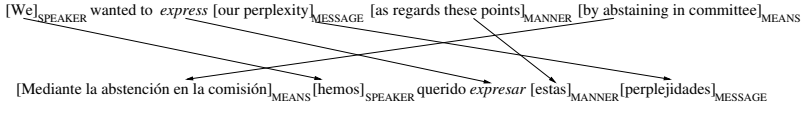

Figure 3.3: *An example of automatic markup and transfer of FEs and target in a sentence from the European Parliament corpus.*

Figure 3.3 shows an automatically annotated sentence in English and its counterpart in Spanish. The example demonstrates the two possible sources of errors: first, incorrect English annotation (the MANNER role, caused by a prepositional phrase attachment error made by the parser); secondly, errors due to the transfer of bracketing. We have two examples of the second category of errors: first, *we* is mapped onto the auxiliary verb *hemos* 'we have', which is a trivial and regularly appear-

ing error; secondly, English *these* is mapped onto Spanish *estas* 'these', which illustrates a a more fundamental problem of the method that arises due to the fact that the sentences are not literal translations and do not express exactly identical information.

3.5 Results

We evaluated the system for three cases: transfer of target word bracketing, transfer of FE bracketing, and the complete system (i.e. application of the English SRL system, followed by transfer of bracketing). For all cases, evaluation was done by manual inspection. We ignored punctuation and articles when checking the boundaries.

In some cases, there was no Spanish counterpart for an entity on the English side. For FEs, the most common reason for this is that Spanish does not have a mandatory subject pronoun as in English (as in the two figures above). In addition, since the translations are not literal and sentence-by-sentence, the target sentence may not express exactly the same information. In the tables below, these cases are listed as N/A.

3.5.1 Target Word Bracketing Transfer

We first measured how well target word bracketing was projected across languages. Table 3.1 shows the results. Since the amount of available text is rather large, and since we perform no FE transfer for target words that can't be transferred, precision is more important than recall for this task in order to produce a high-quality annotation.

Correct transfer	102
N/A	53
Overlapping	23
No overlap	9
Lost in transfer	40
Noise	13
Precision	0.69
Recall	0.59

Table 3.1: *Results of target word transfer.*

Spurious target words were sometimes a problem, especially for the verbs *take* and *make*, which seem to occur as a part of multiword units,

or as support verbs for noun predicatives, more often than in their concrete sense. Disambiguating these uses is a very complex problem that we could not address. When such words were transferred, they were listed as "noise" in Table 3.1. This problem could often be side-stepped, since the word aligner frequently found no counterparts of these words on the Spanish side.

Word sense ambiguity of the target word was a frequent problem. FrameNet often does not cover all senses of a target word (sometimes not even the most common one). We did not have time to try the recent 1.2 release of FrameNet, but we expect that the issue of sense coverage will be less of a problem in the new release. The FrameNet annotators state that the new release has been influenced by their recent annotation of running text.

3.5.2 FE Bracketing Transfer

Table 3.2 shows the results of the transfer of FE annotation for those sentences where a projected target word had been found.

Correct transfer	129
N/A	33
Pronoun to auxiliary	7
Overlapping	9
Lost in transfer	5
No overlap	16
Precision	0.84
Recall	0.81

Table 3.2: *Results of FE transfer for sentences with non-empty targets.*

A few errors were caused by the alignment of English personal pronouns with Spanish auxiliary verbs (such as in Figure 3.3). These cases are listed as "pronoun to auxiliary" in Table 3.2. Since these cases are restricted and easily detected, we did not include them among the errors when computing precision and recall.

3.5.3 Full Annotation

We finally made a tentative study on how well the final result turned out. Since we lacked the lexical expertise to produce a annotated gold standard in the short time span available, we manually inspected the

FEs and labeled them as Acceptable or not. This allowed us to measure the precision of the annotation process, but we did not attempt to measure the recall since we had no gold standard. Because of the sometimes subtle differences between different frames and different semantic roles in the same frame, the result may be somewhat inexact.

Table 3.3 shows the results of the complete semantic role labeling process for sentences where a target word was found on the Spanish side. We have not attempted to label null-instantiated roles.

Acceptable label and boundaries	98
N/A	33
Pronoun to auxiliary	7
Acceptable label, overlapping	12
Incorrect label or no overlap	44
Precision	0.64

Table 3.3: *Results of complete semantic role labeling for sentences with non-empty targets.*

The precision (0.64) is consistent with the result on the FrameNet test set (0.72) multiplied by the transfer precision (0.84), which gives the result 0.60. Extrapolating this argument to the case of recall, we would arrive at a result of $0.63 \cdot 0.81 = 0.51$. However, this figure is probably too high, since there will be FEs on the Spanish side that have no counterpart on the English side.

3.6 Conclusion and Future Work

We have described a method for projection of FrameNet annotation across languages using parallel corpora and a word aligner. Although the produced data for obvious reasons have inferior quality compared to manually produced data, they can be used as a seed for bootstrapping methods, as argued by Padó and Lapata (2005a). In addition, we believe that the method is fully usable in a semi-automatic system. Inspection and correction is less costly than manual annotation from scratch.

We will try to improve the robustness of the methods. Since most sentences in the Parliament debates are long and structurally complicated, it might be possible to improve the data quality by selecting shorter sentences. This should make the task easier for the parser, the

English SRL system, and the word aligner. Parse and alignment probability scores could also be used for selection of data of good quality.

We will create a gold standard in order to be able to estimate the recall, and get more reliable figures for the precision.

Frame assignment is still lacking. We believe that this is best solved using a joint optimization of frame and role assignment as by Thompson et al. (2003), or possibly by applying Lesk's algorithm (Lesk, 1986) using the frame definitions. Other aspects of FrameNet annotation that should be addressed include aspectual particles of verbs and support verbs and prepositions. Our English SRL system partly addresses this task (Johansson and Nugues, 2006a), but it still needs to be exploited to improve the projection methods.

We will further investigate the projection methods to see if a more sophisticated approach than the maximal span method can be applied. Although our method, which is based on the alignment of raw words, is independent of language, it would be interesting to study if the results could be improved if morpheme information is used. In addition, we would like to study if the boundaries of the projected arguments may be adjusted using a parser or chunker.

In the future, we will apply this method for other kinds of semantic annotation of text. Important examples of this is TimeML annotation of events and temporal relations (Pustejovsky et al., 2003a, 2005) and annotation of coreference relations. However, those types of data may be less suitable for automatic projection methods since larger pieces of text than sentences would have to be automatically aligned.

Chapter 4

A FrameNet-based Semantic Role Labeler for Swedish Text

We describe the implementation of a FrameNet-based SRL system for Swedish text. To train the system, we used a semantically annotated corpus that was produced by projection across parallel corpora. As part of the system, we developed two frame element bracketing algorithms that are suitable when no robust constituent parsers are available.

Apart from being the first such system for Swedish, this is as far as we are aware the first semantic role labeling system for a language for which no role-semantic annotated corpora are available.

The estimated accuracy of classification of pre-segmented frame elements is 0.75, and the precision and recall measures for the complete task are 0.67 and 0.47, respectively.

4.1 Introduction

Automatic extraction and labeling of semantic arguments of predicates, or *semantic role labeling* (SRL), has been an active area of research during the last few years. SRL systems have proven useful in a number of NLP projects. The main reason for their popularity is that they can produce a flat layer of semantic structure with a fair degree of robustness.

Building SRL systems for English has been studied widely (Gildea and Jurafsky, 2002; Litkowski, 2004; Carreras and Màrquez, 2005). Most

of them are based on the FrameNet (Baker et al., 1998) or PropBank (Palmer et al., 2005) annotation standards. However, all these works rely on corpora that have been produced at the cost of an enormous effort by human annotators. The current FrameNet corpus, for instance, consists of 130,000 manually annotated sentences. For smaller languages such as Swedish, such corpora are not available.

In this work, we used an English-Swedish parallel corpus whose English part was annotated with semantic roles using the FrameNet standard. We then applied a *cross-language transfer* to derive an annotated Swedish part. This annotated corpus was used to train a complete semantic role labeler for Swedish. We evaluated the system by applying it to a small portion of the FrameNet example corpus that was translated manually.

Cross-language projection of linguistic annotation has been used for a few years, for example to create chunkers and named entity recognizers (Yarowsky et al., 2001) or parsers (Hwa et al., 2002) for languages for which annotated corpora are scarce. Recently, as seen in Chapter 3, these methods have been applied to FrameNet annotation.

4.2 Automatic Annotation of a Swedish Training Corpus

4.2.1 Training an English Semantic Role Labeler

We selected the 150 most common frames in FrameNet and applied the Collins parser (Collins, 1999) to the example sentences for those frames. We built a conventional FrameNet parser for English using 100,000 of those sentences as a training set and 8,000 as a development set. The classifiers were based on SVMs that we trained using LIBSVM (Chang and Lin, 2001) with the Gaussian kernel. When testing the system, we did not assume that the frame was known a priori. We used the available semantic roles for all senses of the target word as features for the classifier. See Johansson and Nugues (2006a) for more details.

On a test set from FrameNet, we estimated that the system had a precision of 0.71 and a recall of 0.65 using a strict scoring method. The result is slightly lower than the best systems at Senseval-3 (Litkowski, 2004), possibly because we used a larger set of frames, and we did not assume that the frame was known a priori.

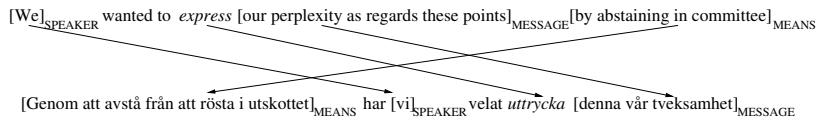

Figure 4.1: *Example of projection of FrameNet annotation.*

4.2.2 Transferring the Annotation

We produced a Swedish-language corpus annotated with FrameNet information by applying the SRL system to the English side of Europarl (Koehn, 2005), which is a parallel corpus that is derived from the proceedings of the European Parliament. We projected the bracketing of the target words and the frame elements onto the Swedish side of the corpus by using the Giza++ word aligner (Och and Ney, 2003). Each word on the English side was mapped by the aligner onto a (possibly empty) set of words on the Swedish side. We used the maximal span method to infer the bracketing on the Swedish side, which means that the span of a projected entity was set to the range from the leftmost projected token to the rightmost. Figure 4.1 shows an example of this process.

To make the brackets conform to the FrameNet annotation standard, we applied a small set of heuristics. The FrameNet conventions specify that linking words such as prepositions and subordinating conjunctions should be included in the bracketing. However, since constructions are not isomorphic in the sentence pair, a linking word on the target side may be missed by the projection method since it is not present on the source side. For example, the sentence *the doctor was answering an emergency phone call* is translated into Swedish as *doktorn svarade **på** ett larmsamtal*, which uses a construction with a preposition *på* 'to/at/on' that has no counterpart in the English sentence. The heuristics that we used are specific for Swedish, although they would probably be very similar for any other language that uses a similar set of prepositions and connectives, i.e. most European languages.

We used the following heuristics:

- When there was only a linking word (preposition, subordinating conjunction, or infinitive marker) between the FE and the target word, it was merged with the FE.
- When a Swedish FE was preceded by a linking word, and the English FE starts with such a word, it was merged with the FE.

- The FE brackets were adjusted to the output of a chunker to include only complete chunks.
- When a Swedish FE crossed the target word, we used only the part of the FE that was on the right side of the target.

In addition, some bad annotation was discarded because we obviously could not use sentences where no counterpart for the target word could be found. Additionally, we used only those sentence where the target word was mapped to a noun, verb, or an adjective on the Swedish side.

Because of homonymy and polysemy problems, applying a SRL system without knowing target words and frames a priori necessarily introduces noise into the automatically created training corpus. There are two kinds of word sense ambiguity that are problematic in this case: the "internal" ambiguity, or the fact that there may be more than one frame for a given target word; and the "external" ambiguity, where frequently occurring word senses are not listed in FrameNet. To sidestep the problem of internal ambiguity, we used the available semantic roles for all senses of the target word as features for the classifier (as described above). Solving the problem of external ambiguity was outside the scope of this work.

Some potential target words had to be ignored since their sense ambiguity was too difficult to overcome. This category includes auxiliaries such as *be* and *have*, as well as verbs such as *take* and *make*, which frequently appear as support verbs for nominal predicates.

4.3 Training a Swedish SRL System

Using the transferred FrameNet annotation, we trained a SRL system for Swedish text. Like most previous systems, it consists of two parts: a FE bracketer and a classifier that assigns semantic roles to FEs. Both parts are implemented as SVM classifiers trained using LIBSVM. The semantic role classifier is rather conventional and is not described in this paper.

To construct the features used by the classifiers, we used the following tools:

- An HMM-based POS tagger (Carlberger and Kann, 1999),
- A rule-based chunker that brackets noun, verb, adjective, prepositional, and adverb groups,

- A rule-based time expression detector (Berglund, 2004),
- Two clause identifiers, of which one is rule-based (Ejerhed, 1996) and one is statistical (Johansson, 2005),
- The MALTPARSER dependency parser (Nivre et al., 2004), trained on Talbanken (Einarsson, 1976), a 100,000-word Swedish treebank.

We constructed shallow parse trees using the clause trees and the chunks. Dependency and shallow parse trees for a fragment of a sentence from our test corpus are shown in Figures 4.2 and 4.3, respectively. This sentence comes from the English FrameNet example corpus and has been manually translated into Swedish. In English, the fragment was *the doctor was answering an emergency phone call*.

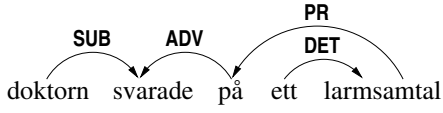

Figure 4.2: *Example dependency parse tree.*

[[doktorn]$_{NG\_nom}$ [svarade]$_{VG\_fin}$ [på]$_{PP}$ [ett larmsamtal]$_{NG\_nom}$]$_{Clause}$

Figure 4.3: *Example shallow parse tree.*

4.3.1 Frame Element Bracketing Methods

We created two FE bracketing algorithms based on binary classification of chunks as starting or ending the FE. This is somewhat similar to the chunk-based system described by Pradhan et al. (2005a), which uses a segmentation strategy based on IOB2 bracketing. However, our system still exploits the dependency parse tree during classification.

We first tried the conventional approach to the problem of FE bracketing: applying a parser to the sentence, and classifying each node in the parse tree as being an FE or not. We used a dependency parser since there is no constituent-based parser available for Swedish. This proved unsuccessful because the spans of the dependency subtrees frequently were incompatible with the FrameNet annotation standard. This was

especially the case for non-verbal target words and when the head of the argument was above the target word in the dependency tree. To be usable, this approach would require some sort of transformation, possibly a conversion into a phrase-structure tree, to be applied to the dependency trees to align the spans with the FEs. Preliminary investigations were unsuccessful, and we leave this to future work.

We believe that the methods we developed are more suitable in our case, since they base their decisions on several parse trees (in our case, two clause-chunk trees and one dependency tree). This redundancy is valuable because the dependency parsing model was trained on a treebank of just 100,000 words, which makes it less robust than Collins' or Charniak's parsers for English. Recent work in semantic role labeling (see for example Pradhan et al., 2005b) has focused on combining the results of SRL systems based on different types of syntax. Still, all systems exploiting recursive parse trees are based on binary classification of nodes as being an argument or not.

The training sets used to train the final classifiers consisted of one million training instances for the start classifier, 500,000 for the end classifier, and 272,000 for the role classifier. The features used by the classifiers are described in Subsection 4.3.2, and the performance of the two FE bracketing algorithms compared in Subsection 4.4.2.

Greedy start-end

The first FE bracketing algorithm, the *greedy start-end* method, proceeds through the sequence of chunks in one pass from left to right. For each chunk opening bracket, a binary classifier decides if an FE starts there or not. Similarly, another binary classifier tests chunk end brackets for ends of FEs. To ensure compliance to the FrameNet annotation standard (bracket matching, and no FE crossing the target word), the algorithm inserts additional end brackets where appropriate. Pseudocode is given in Algorithm 1.

Figure 4.4 shows an example of this algorithm, applied to the example fragment. The small brackets correspond to chunk boundaries, and the large brackets to FE boundaries that the algorithm inserts. In the example, the algorithm inserts an end bracket after the word *doktorn* 'the doctor', since no end bracket was found before the target word *svarade* 'was answering'.

Algorithm 1 Greedy Bracketing

Input: A list L of chunks and a target word t
Binary classifiers `starts` and `ends`
Output: The sets S and E of start and end brackets
Split L into the sublists L_{before}, L_{target}, and L_{after}, which correspond to the parts of the list that is before, at, and after the target word, respectively.
Initialize `chunk-open` to FALSE
for L_{sub} **in** $\{L_{\text{before}}, L_{\text{target}}, L_{\text{after}}\}$ **do**
  for c **in** L_{sub} **do**
    if `starts`(c) **then**
      if `chunk-open` **then**
        Add an end bracket before c to E
      end if
      `chunk-open` $\leftarrow$ TRUE
      Add a start bracket before c to S
    end if
    if `chunk-open` $\wedge$ (`ends`$(c) \vee c$ is final in L_{sub}) **then**
      `chunk-open` $\leftarrow$ FALSE
      Add an end bracket after c to E
    end if
  end for
end for

Globally optimized start-end

The second algorithm, the *globally optimized start-end* method, maximizes a global probability score over each sentence. For each chunk opening and closing bracket, probability models assign the probability of an FE starting (or ending, respectively) at that chunk. The probabilities are estimated using the built-in sigmoid fitting methods of LIBSVM. Making the somewhat unrealistic assumption of independence of the brackets, the global probability score to maximize is defined as the product of all start and end probabilities. We added a set of constraints to ensure that the segmentation conforms to the FrameNet annotation standard. The constrained optimization problem is then solved using the JaCoP finite domain constraint solver (Kuchcinski, 2003). We believe that an n-best beam search method would produce similar results. The pseudocode for the method can be seen in Algorithm 2. The definitions of the predicates `no-nesting` and `no-crossing`, which should

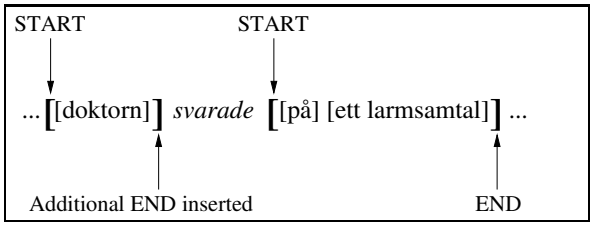

Figure 4.4: *Illustration of the greedy start-end method.*

be obvious, are omitted.

Algorithm 2 Globally Optimized Bracketing

Input: A list L of chunks and a target word t
Probability models $\hat{P}_{\text{starts}}$ and $\hat{P}_{\text{ends}}$
Output: The sets $S_{\max}$ and $E_{\max}$ of start and end brackets

$$\begin{aligned}
\texttt{legal}(S,E) \quad &\leftarrow \quad |S| = |E| \\
&\wedge \quad \max(E) > \max(S) \wedge \min(S) < \min(E) \\
&\wedge \quad \texttt{no-nesting}(S,E) \wedge \texttt{no-crossing}(t,S,E) \\
\texttt{score}(S,E) \quad &\leftarrow \quad \textstyle\prod_{c\in S} \hat{P}_{\text{starts}}(c) \cdot \prod_{c\in L\setminus S}(1-\hat{P}_{\text{starts}}(c)) \\
&\quad\quad \cdot \textstyle\prod_{c\in E} \hat{P}_{\text{ends}}(c) \cdot \prod_{c\in L\setminus E}(1-\hat{P}_{\text{ends}}(c))
\end{aligned}$$

$$(S_{\max}, E_{\max}) \leftarrow \text{argmax}_{\{\texttt{legal}(S,E)\}} \texttt{score}(S,E)$$

Figure 4.5 shows an example of the globally optimized start-end method. In the example, the global probability score is maximized by a bracketing that is illegal because the FE starting at *doktorn* is not closed before the target ($0.8 \cdot 0.6 \cdot 0.6 \cdot 0.7 \cdot 0.8 \cdot 0.7 = 0.11$). The solution of the constrained problem is a bracketing that contains an end bracket before the target ($0.8 \cdot 0.4 \cdot 0.6 \cdot 0.7 \cdot 0.8 \cdot 0.7 = 0.075$)

4.3.2 Features Used by the Classifiers

Table 4.1 summarizes the feature sets used by the greedy start-end (GSE), optimized start-end (OSE), and semantic role classification (SRC).

Conventional Features

Some of the features we use are well-known from literature. Most of them have been used by almost every system since the first well-known

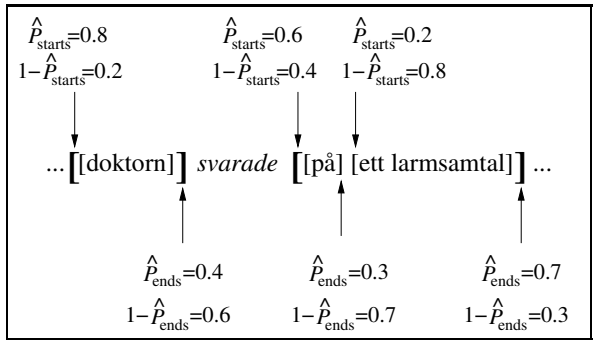

Figure 4.5: *Illustration of the globally optimized start-end method.*

description (Gildea and Jurafsky, 2002). These features are used by all classifiers:

- *Target word (predicate) lemma and POS*
- *Voice* (when the target word is a verb)
- *Position* (before or after the target)
- *Head word and POS*
- *Phrase or chunk type*

In addition, all classifiers use the set of allowed semantic role labels as a set of boolean features. This is needed to constrain the output to a label that is allowed by FrameNet for the current frame. In addition, this feature has proven useful for the FE bracketing classifiers to distinguish between event-type and object-type frames. For event-type frames, dependencies are often long-distance, while for object-type frames, they are typically restricted to chunks very near the target word. The part of speech of the target word alone is not enough to distinguish these two classes, since many nouns belong to event-type frames.

For the phrase/chunk type feature, we use slightly different values for the bracketing case and the role assignment case: for bracketing, the value of this feature is simply the type of the current chunk; for classification, it is the type of the largest chunk or clause that starts at the leftmost token of the FE. For prepositional phrases, the preposition is attached to the phrase type (for example, the second FE in the example fragment starts with the preposition *på* 'at/on', which causes the value of the phrase type feature to be PP-*på*).

	GSE	OSE	SRC
Target lemma	+	+	+
Target POS	+	+	+
Voice	+	+	+
Allowed role labels	+	+	+
Position	+	+	+
Head word (HW)	+	+	+
Head POS	+	+	+
Phrase/chunk type (PT)	+	+	+
HW/POS/PT, $\pm$2 chunk window	+	+	-
Dep-tree & shallow path $\rightarrow$target	+	+	+
Starting paths $\rightarrow$target	+	+	-
Ending paths $\rightarrow$target	+	+	-
Path $\rightarrow$start	+	-	-

Table 4.1: *Features used by the classifiers.*

Chunk Context Features

Similarly to the chunk-based PropBank argument bracketer described by Pradhan et al. (2005a), the start-end methods use the head word, head POS, and chunk type of chunks in a window of size 2 on both sides of the current chunk to classify it as being the start or end of an FE.

Parse Tree Path Features

Parse tree path features have been shown to be very important for argument bracketing in several studies (Gildea and Palmer, 2002; Punyakanok et al., 2005). All classifiers used here use a set of such features:

- *Dependency tree path from the head to the target word.* In the example text, the first chunk (consisting of the word *doktorn*), has the value SUB-↑ for this feature. This means that to go from the head of the chunk to the target in the dependency graph (Figure 4.2), you traverse a SUB (subject) link upwards. Similarly, the last chunk (*ett larmsamtal*) has the value PR-↑-ADV-↑.

- *Shallow path from the chunk containing the head to the target word.* For the same chunks as above, these values are both NG_nom-↑-Clause-↓-VG_fin, which means that to traverse the shallow parse

tree (Figure 4.3) from the chunk to the target, you start with a NG_nom node, go upwards to a Clause node, and finally down to the VG_fin node.

The start-end classifiers additionally use the full set of paths (dependency and shallow paths) to the target word from each node starting (or ending, respectively) at the current chunk, and the greedy end classifier also uses the path from the current chunk to the start chunk.

4.4 Evaluation of the System

4.4.1 Evaluation Corpus

To evaluate the system, we manually translated 150 sentences from the FrameNet example corpus. These sentences were selected randomly from the English development set. Some sentences were removed, typically because we found the annotation dubious or the meaning of the sentence difficult to comprehend precisely. The translation was mostly straightforward. Because of the extensive use of compounding in Swedish, some frame elements were merged with target words.

4.4.2 Comparison of FE Bracketing Methods

We compared the performance of the two methods for FE bracketing on the test set. Because of limited time, we used smaller training sets than for the full evaluation below (100,000 training instances for all classifiers). Table 4.2 shows the result of this comparison.

	Greedy	Optimized
Precision	0.70	0.76
Recall	0.50	0.44
$F_{\beta=1}$	0.58	0.55

Table 4.2: *Comparison of FE bracketing methods.*

As we can see from the Table 4.2, the globally optimized start-end method increased the precision somewhat, but decreased the recall and made the overall F-measure lower. We therefore used the greedy start-end method for our final evaluation that is described in the next section.

4.4.3 Final System Performance

We applied the Swedish semantic role labeler to the translated sentences and evaluated the result. We used the conventional experimental setting where the frame and the target word were given in advance. The results, with approximate 95% confidence intervals included, are presented in Table 4.3. The figures are precision and recall for the full task, classification accuracy of pre-segmented arguments, precision and recall for the bracketing task, full task precision and recall using the Senseval-3 scoring metrics, and finally the proportion of full sentences whose FEs were correctly bracketed and classified. The Senseval-3 method uses a more lenient scoring scheme that counts a FE as correctly identified if it overlaps with the gold standard FE and has the correct label. Although the strict measures are most interesting, we include these figures for comparison with the systems participating in the Senseval-3 Restricted task (Litkowski, 2004).

We include baseline scores for the argument bracketing and classification tasks, respectively. For the bracketing case, the baseline selects the words in each subtree[1] that is a dependent of the predicate node in the dependency parse tree when the predicate is a verb. When it is a noun, we also add the predicate token itself, and when it is an adjective, we additionally add the parent token of the predicate node. For the argument classifier, the baseline selects the most frequent semantic role in that frame for each argument. As can be seen from the table, all scores except the argument bracketing recall are well above the baselines.

Although the performance figures are better than the baselines, they are still lower than for most English systems (although higher than some of the systems at Senseval-3). We believe that the main reason for the performance is the quality of the data that were used to train the system, since the results are consistent with the hypothesis brought forward in Chapter 3: that the quality of the transferred data was roughly equal to the performance of the English system multiplied by the figures for the transfer method. In that experiment, the transfer method had a precision of 0.84, a recall of 0.81, and an F-measure of 0.82. If we assume that the transfer performance is similar for Swedish, we arrive at a precision of $0.71 \cdot 0.84 = 0.60$, a recall of $0.65 \cdot 0.81 = 0.53$, and an F-measure of 0.56. For the F-measure, 0.55 for the system and 0.56 for the product, the figures match closely. For the precision, the system performance (0.67) is significantly higher than the product (0.60), which

[1] This is possible because MALTPARSER produces projective trees, i.e. the words in each subtree form a contiguous string.

Precision (Strict scoring method)	0.67 ± 0.064
Recall	0.47 ± 0.057
Argument Classification Accuracy	0.75 ± 0.050
Baseline	0.41 ± 0.056
Argument Bracketing Precision	0.80 ± 0.055
Baseline Precision	0.50 ± 0.055
Argument Bracketing Recall	0.57 ± 0.057
Baseline Recall	0.55 ± 0.057
Precision (Senseval-3 scoring method)	0.77 ± 0.057
Overlap	0.75 ± 0.039
Recall	0.55 ± 0.057
Complete Sentence Accuracy	0.29 ± 0.073

Table 4.3: *Results on the Swedish test set with approximate 95% confidence intervals.*

suggests that the SVM learning method handles the noisy training set rather well for this task. The recall (0.47) is lower than the corresponding product (0.53), but the difference is not statistically significant at the 95% level. These figures suggest that the main effort towards improving the system should be spent on improving the training data.

4.5 Conclusion

We have described the design and implementation of a Swedish FrameNet-based SRL system that was trained using a corpus that was annotated using cross-language transfer from English to Swedish. With no manual effort except for translating sentences for evaluation, we were able to reach promising results. To our knowledge, the system is the first SRL system for Swedish in literature. As long as there is a parallel corpus (where one of the languages is English) available, we believe that the methods described could be applied to any language, although the relatively close relationship between English and Swedish probably made the task comparatively easy in our case.

However, as we can see, the figures (especially the FE bracketing recall) leave room for improvement. Apart from the noisy training set, probable reasons for this include the lower robustness of the Swedish parsers compared to those available for English. In addition, we have noticed that the European Parliament corpus is somewhat biased. For

instance, a very large proportion of the target words evoke the STATEMENT or DISCUSSION frames, but there are very few instances of the BEING_WET and MAKING_FACES frames. While training, we tried to balance the selection somewhat, but applying the projection methods on other type of parallel corpora (such as novels available in both languages) may produce a better training corpus.

Chapter 5

Conclusion and Future Work

This thesis has described an architecture and a prototype implementation of a system that automatically converts texts describing physical processes into animated images. The prototype system, Carsim, was restricted to traffic accident reports written in Swedish, but we think that the architecture can be ported to other languages and other domains. We provided an evaluation of the system, and although the figures are somewhat low for a fully automatic system, we believe that they are perfectly acceptable in a semi-automatic system that forms a part of the tool chain used by media producers. Such a system may speed up the process of graphics production.

As a first step towards generalization of the semantic components of the Carsim NLP module, we developed a FrameNet-based semantic role labeler for Swedish. We believe that this is the first time such a system is produced without the use of manually annotated corpora. The results are promising, but the argument identification recall needs to be improved for the system to be fully usable in practice. To this end, we will try to develop methods to improve the quality of the automatically produced training data or devise heuristics to select data of good quality. Additionally, we will investigate whether a wider range of syntactic resources, such as additional parsers or suitable transformations of the parse trees, improves the performance.

We believe, and a few studies suggest, that an automatic system for identification and classification of predicate arguments may be useful for the problem of open-domain information extraction in general —

not just for the specific purpose of automatic illustration. However, the mapping from predicate/argument structures to the output format of the information extraction system may be complex, such as in the Carsim system. In Carsim and all other systems described in literature, this mapping is carried out by applying hand-written rules. We would like to study if it could instead be replaced by an automatic system, which would reduce development time greatly. However, this may be impeded by very complex reference problems which may be impossible to solve in the general case (such as underspecification and metonymy), at least when the mapping consists of something more than pasting text snippets into template slots.

The problem of automatic production of training data was the main focus of the latter part of this thesis. While we presented a method to automatically produce annotated data by using aligned sentences, which works fine in the case of FrameNet, it still remains to be seen if similar alignment and projection methods can be used on structures that run over complete texts rather than single sentences or even clauses. This will be necessary for annotation of coreference relations or temporal structure to be transferred to a new language, for instance.

A crucial task for a conversion of a text into an animated sequence is to automatically determine the temporal structure of the sequence of events. Carsim includes a module that works well for traffic accident reports, but preliminary experiments suggest that this problem is much more difficult to solve for open domains and longer texts, such as those included in the TimeBank corpus. This task may need to involve deeper representations of discourse structure and larger quantities of lexical and real-world knowledge to be solvable in a less restricted case.

We may conclude by saying that automatic illustration of sequences of events has been proven possible by this study, but that the goals set must be modest for such a system to be successful. When moving to a more general setting from a limited domain, we run into very deep linguistic problems that cannot be expected to be solved in the near future, if ever. However, the success in one restricted case may probably be replicated in other restricted cases. By designing generic components that are easily portable to new domains, such a transition may hopefully proceed smoothly.

Bibliography

Adorni, G., Manzo, M. D., and Giunchiglia, F. (1984). Natural language driven image generation. In *Proceedings of COLING 84*, pages 495–500, Stanford, California.

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.

Arens, M., Ottlik, A., and Nagel, H.-H. (2002). Natural language texts for a cognitive vision system. In van Harmelen, F., editor, *ECAI2002, Proceedings of the 15th European Conference on Artificial Intelligence*, Lyon.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of COLING-ACL'98*, pages 86–90, Montréal, Canada.

Banko, M. and Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In *Meeting of the Association for Computational Linguistics*, pages 26–33.

Bennett, B. (2002). Physical objects, identity and vagueness. In Fensel, D., McGuinness, D., and Williams, M.-A., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Eighth International Conference (KR2002)*, San Francisco, CA. Morgan Kaufmann.

Bennett, B. and Galton, A. P. (2004). A unifying semantics for time and events. *Artificial Intelligence*, 153:13–48.

Berglund, A. (2004). Extracting temporal information and ordering events for Swedish. Master's thesis, Lunds Tekniska Högskola.

Berglund, A., Johansson, R., and Nugues, P. (2006a). Extraction of temporal information from texts in Swedish. In *Proceedings of LREC-2006 (to appear)*, Genoa, Italy.

Berglund, A., Johansson, R., and Nugues, P. (2006b). A machine learning approach to extract temporal information from texts in Swedish and generate animated 3D scenes. In *Proceedings of EACL-2006 (to appear)*, Trento, Italy.

Boas, H. C. (2002). Bilingual FrameNet dictionaries for machine translation. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, volume IV, pages 1364–1371, Las Palmas, Spain.

Boguraev, B. and Ando, R. (2005). TimeML-compiliant text analysis for temporal reasoning. In *Proceedings of IJCAI-2005*, pages 997–1003, Edinburgh, United Kingdom.

Carlberger, J. and Kann, V. (1999). Implementing an efficient part-of-speech tagger. *Software Practice and Experience*, 29:815–832.

Carreras, X. and Màrquez, L. (2005). Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Chang, C.-C. and Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*.

Chen, B. and Fung, P. (2004). Automatic construction of an English-Chinese bilingual FrameNet. In *Proceedings of HLT/NAACL-2004*.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46.

Cohn, A. G. and Hazarika, S. M. (2001). Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46(1-2):1–29.

Collins, M. J. (1999). Head-driven statistical models for natural language parsing. Ph.D. thesis, University of Pennsylvania, Philadelphia.

Coyne, B. and Sproat, R. (2001). WordsEye: An automatic text-to-scene conversion system. In *Proceedings of the Siggraph Conference*, Los Angeles.

Danielsson, M. (2005). Maskininlärningsbaserad koreferensbestämning för nominalfraser applicerat på svenska texter. Master's thesis, Lunds Universitet.

Diab, M. and Resnik, P. (2002). An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*.

Dupuy, S., Egges, A., Legendre, V., and Nugues, P. (2001). Generating a 3D simulation of a car accident from a written description in natural language: The Carsim system. In *Proceedings of The Workshop on Temporal and Spatial Information Processing*, pages 1–8, Toulouse. Association for Computational Linguistics.

Einarsson, J. (1976). Talbankens skriftspråkskonkordans. Department of Scandinavian Languages, Lund University.

Ejerhed, E. (1996). Finite state segmentation of discourse into clauses. In *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI-96) Workshop on Extended Finite State Models of Language*, Budapest, Hungary.

Fillmore, C. J. (1968). The case for case. In Bach, E. R. and Harms, R. T., editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart, and Winston.

Fillmore, C. J. (1976). Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language*, 280:20–32.

Fung, P. and Chen, B. (2004). BiFrameNet: Bilingual frame semantics resource construction by cross-lingual induction. In *Proceedings of Coling-2004*.

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

Gildea, D. and Palmer, M. (2002). The necessity of syntactic parsing for predicate argument recognition. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*, pages 239–246.

Hirschman, L. (1997). MUC-7 coreference task definition, version 3.0. In *Proceedings of MUC-7*.

Hwa, R., Resnik, P., Weinberg, A., and Kolak, O. (2002). Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*.

Johansson, R. (2005). A language-independent statistical clause bracketer. Technical report, Lunds Tekniska Högskola.

Johansson, R., Berglund, A., Danielsson, M., and Nugues, P. (2005). Automatic text-to-scene conversion in the traffic accident domain. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 1073–1078, Edinburgh, Scotland.

Johansson, R. and Nugues, P. (2005a). Automatic conversion of traffic accident reports into 3D animations. In *SIGRAD-05*, Lund, Sweden.

Johansson, R. and Nugues, P. (2005b). Using parallel corpora for automatic transfer of FrameNet annotation. In *Proceedings of the 1st ROMANCE FrameNet Workshop*, Cluj-Napoca, Romania.

Johansson, R. and Nugues, P. (2006a). Automatic annotation for all semantic layers in FrameNet. In *Proceedings of EACL-2006 (to appear)*, Trento, Italy.

Johansson, R. and Nugues, P. (2006b). Construction of a FrameNet labeler for Swedish text. In *Proceedings of LREC-2006 (to appear)*, Genoa, Italy.

Johansson, R., Williams, D., Berglund, A., and Nugues, P. (2004). Carsim: A System to Visualize Written Road Accident Reports as Animated 3D Scenes. In Hirst, G. and Nirenburg, S., editors, *ACL2004: Second Workshop on Text Meaning and Interpretation*, pages 57–64, Barcelona, Spain.

Karlberg, N.-O. (2003). Field results from STRADA – a traffic accident data system telling the truth. In *ITS World Congress*, Madrid, Spain.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT Summit 2005*.

Korf, R. E. (1985). Iterative deepening A*: An optimal admissible tree search. *Artificial Intelligence*, 27(1):97–109.

Kuchcinski, K. (2003). Constraints-driven scheduling and resource assignment. *ACM Transactions on Design Automation of Electronic Systems*, 8(3):355–383.

Lesk, M. E. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 SIGDOC Conference*, pages 24–26.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.

Litkowski, K. (2004). Senseval-3 task: Automatic labeling of semantic roles. In Mihalcea, R. and Edmonds, P., editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 9–12, Barcelona, Spain. Association for Computational Linguistics.

Lu, R. and Zhang, S. (2002). *Automatic Generation of Computer Animation*, volume 2160 of *Lecture Notes in Computer Science*. Springer Verlag.

Ma, M. and Mc Kevitt, P. (2003). Semantic representation of events in 3D animation. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, pages 253–281, Tilburg, The Netherlands.

Ma, M. and Mc Kevitt, P. (2004). Visual semantics and ontology of eventive verbs. In *Proceedings of IJCNLP-2004*, pages 187–196.

Mani, I. and Schiffman, B. (2005). Temporally anchoring and ordering events in news. In Pustejovsky, J. and Gaizauskas, R., editors, *Time and Event Recognition in Natural Language*. John Benjamins (to appear).

Markert, K. and Hahn, U. (2002). Metonymies in discourse. *Artificial Intelligence*, 135(1-2):145–198.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., and Grishman, R. (2004). The NomBank project: An interim report. In Meyers, A., editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Moldovan, D., Clark, C., and Harabagiu, S. (2005). Temporal context representation and reasoning. In *Proceedings of IJCAI-2005*, pages 1099–1104, Edinburgh, United Kingdom.

Moschitti, A., Morărescu, P., and Harabagiu, S. (2003). Open domain information extraction via automatic semantic labeling. In *Proceedings of the 2003 Special Track on Recent Advances in Natural Language at the 16th International FLAIRS Conference*, St. Augustine, Florida.

Narayanan, S. and Harabagiu, S. (2004). Question answering based on semantic structures. In *International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

Nivre, J., Hall, J., and Nilsson, J. (2004). Memory-based dependency parsing. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL)*, pages 49–56, Boston.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Padó, S. and Lapata, M. (2005a). Cross-lingual bootstrapping for semantic lexicons: The case of FrameNet. In *Proceedings of AAAI-05*, Pittsburgh, PA.

Padó, S. and Lapata, M. (2005b). Cross-lingual projection of role-semantic information. In *Proceedings of HLT/EMNLP 2005*.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1).

Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J., and Jurafsky, D. (2005a). Support vector learning for semantic argument classification. *Machine Learning*, 60(1):11–39.

Pradhan, S., Ward, W., Hacioglu, K., Martin, J., and Jurafsky, D. (2005b). Semantic role labeling using different syntactic views. In *Proceedings of ACL-2005*.

Punyakanok, V., Roth, D., and Yih, W. (2005). The necessity of syntactic parsing for semantic role labeling. In *Proceedings of IJCAI-2005*.

Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., and Katz, G. (2003a). TimeML: Robust specification of event and temporal expressions in text. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, Tilburg, The Netherlands.

Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., and Lazo, M. (2003b). The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*, pages 647–656, Lancaster, United Kingdom.

Pustejovsky, J., Ingria, R., Saurí, R., Castaño, J., Littman, J., Gaizauskas, R., Setzer, A., Katz, G., and Mani, I. (2005). The specification language TimeML. In Mani, I., Pustejovsky, J., and Gaizauskas, R., editors, *The Language of Time: a Reader*. Oxford University Press.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kauffman.

Riloff, E., Schafer, C., and Yarowsky, D. (2002). Inducing information extraction systems for new languages via cross-language projection. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*.

Saurí, R., Knippen, R., Verhagen, M., and Pustejovsky, J. (2005). Evita: a robust event recognizer for QA systems. In *Proceedings of HLT/EMNLP 2005*, pages 700–707.

Setzer, A. and Gaizauskas, R. (2001). A pilot study on annotating temporal relations in text. In *ACL 2001, Workshop on Temporal and Spatial Information Processing*, pages 73–80, Toulouse, France.

Simmons, R. F. (1975). The CLOWNS microworld. In *TINLAP '75: Proceedings of the 1975 workshop on Theoretical issues in natural language processing*, pages 17–19, Morristown, NJ, USA. Association for Computational Linguistics.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Sproat, R. (2001). Inferring the environment in a text-to-scene conversion system. In *Proceedings of the K-CAP'01*.

Surdeanu, M., Harabagiu, S., Williams, J., and Aarseth, P. (2003). Using predicate-argument structures for information extraction. In *Proceedings of the ACL*, Sapporo, Japan.

Swier, R. S. and Stevenson, S. (2004). Unsupervised semantic role labelling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 95–102, Barcelona, Spain.

Swier, R. S. and Stevenson, S. (2005). Exploiting a verb lexicon in automatic semantic role labelling. In *Proceedings of the Joint Human Language Technology Conference and Conference on Empirical Methods*

in Natural Language Processing (HLT/EMNLP-05), Vancouver, British Columbia.

Thompson, C. A., Levy, R., and Manning, C. (2003). A generative model for FrameNet semantic role labeling. In *Proceedings of the 14th European Conference on Machine Learning*.

van Deemter, K. and Kibble, R. (2000). On coreferring: Coreference annotation in MUC and related schemes. *Computational Linguistics*, 26(4):615–623.

Verhagen, M., Mani, I., Saurí, R., Littman, J., Knippen, R., Jang, S. B., Rumshisky, A., Phillips, J., and Pustejovsky, J. (2005). Automating temporal annotation with TARSQI. In *Proceedings of the ACL 2005*.

Viberg, Å., Lindmark, K., Lindvall, A., and Mellenius, I. (2002). The Swedish WordNet project. In *Proceedings of Euralex 2002*, pages 407–412, Copenhagen.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, MA.

Yarowsky, D., Ngai, G., and Wicentowski, R. (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT 2001, First International Conference on Human Language Technology Research*.

Appendix A

Acronyms

FE Frame Element

IE Information Extraction

NLP Natural Language Processing

SRL Semantic Role Labeling

SVM Support Vector Machine