Abductive Reasoning with a Large Knowledge Base for Discourse Processing

Ekaterina Ovchinnikova, Niloofar Montazeri, Theodore Alexandrov, Jerry R. Hobbs, Michael C. McCord, and Rutu Mulkar-Mehta

1 Introduction

In this chapter, we elaborate on a semantic processing framework based on a mode of inference called *abduction*, or inference to the best explanation. In logic, abduction is a kind of inference which arrives at an explanatory hypothesis given an observation. Hobbs et al. (1993) describe how abductive reasoning can be applied to the discourse processing problem viewing the process of interpreting sentences in discourse as the process of providing the best explanation of why the sentence would be true. In this framework, interpreting a sentence means

- proving its logical form,
- merging redundancies where possible, and
- making assumptions where necessary.

As the reader will see later in this chapter, abductive reasoning as a discourse processing technique helps to solve many pragmatic problems such as reference resolu-

Ekaterina Ovchinnikova
USC ISI, 4676 Admiralty Way Marina del Rey, CA 90292, USA, e-mail: katya@isi.edu

Niloofar Montazeri
USC ISI, 4676 Admiralty Way Marina del Rey, CA 90292, USA, e-mail: niloofar@isi.edu

Theodore Alexandrov
University of Bremen, Bibliothekstr. 1, 28359 Bremen, Germany, e-mail: theodore@uni-bremen.de

Jerry R. Hobbs
USC ISI, 4676 Admiralty Way Marina del Rey, CA 90292, USA, e-mail: hobbs@isi.edu

Michael C. McCord
Independent Researcher, e-mail: mcmccord@member.ams.org

Rutu Mulkar-Mehta
UCSD-SDSC, 9500 Gilman Dr., La Jolla, California 92093-0505, USA, e-mail: me@rutumulkar.com

tion, the interpretation of noun compounds, and the resolution of some kinds of syntactic and semantic ambiguity as a by-product. We adopt this approach. Specifically, we use a system we have built called *Mini-TACITUS*[1] (Mulkar et al., 2007) that provides the expressivity of logical inference but also allows probabilistic, fuzzy, or defeasible inference and includes measures of the "goodness" of abductive proofs and hence of interpretations of texts and other situations.

The success of a discourse processing system based on inferences heavily depends on a knowledge base. This chapter shows how a large and reliable knowledge base can be obtained by exploiting existing lexical semantic resources and can be successfully applied to reasoning tasks on a large scale. In particular, we experiment with axioms extracted from WordNet (Fellbaum, 1998), and FrameNet (Ruppenhofer et al., 2006). In axiomatizing FrameNet we rely on the study described in (Ovchinnikova et a., 2010; Ovchinnikova, 2012).

We evaluate our inference system and knowledge base in recognizing textual entailment (RTE). As the reader will see in the following sections, inferences carried out by *Mini-TACITUS* are fairly general and not tuned for a particular application. We decided to test our approach on RTE because this is a well-defined task that captures major semantic inference needs across many natural language processing applications, such as question answering, information retrieval, information extraction, and document summarization. For evaluation, we have chosen the RTE-2 Challenge data set (Bar-Haim et al., 2006), because besides providing text-hypothesis pairs and a gold standard this data set has been annotated with FrameNet frame and role labels (Burchardt and Pennacchiotti, 2008), which gives us the possibility of evaluating our frame and role labeling based on the axioms extracted from FrameNet.

This chapter is structured as follows. Section 2 introduces weighted abduction. In section 3, we briefly describe our discourse processing pipeline and explain how abductive reasoning can be applied to discourse processing. Section 4 concerns unification in weighted abduction. In section 5, we describe the obtained knowledge base. In section 6, optimizations of the *Mini-TACITUS* system required to make the system able to handle large knowledge bases are described. Section 7 presents our procedure for recognizing textual entailment. In section 8, we provide an evaluation of our discourse processing pipeline on the RTE-2 data set. The last section concludes the chapter and gives an outlook on future work and perspectives.

2 Weighted Abduction

Abduction is inference to the best explanation. Formally, logical abduction is defined as follows:

[1] http://www.rutumulkar.com/tacitus.html

Given: Background knowledge B, observations O, where both B and O are sets of first-order logical formulas,

Find: A hypothesis H such that $H \cup B \models O, H \cup B \not\models \bot$, where H is a set of first-order logical formulas.

Typically, there exist several hypotheses H explaining O. To rank candidate hypotheses according to plausibility, we use the framework of *weighted abduction* as defined by Hobbs et al. (1993). In this framework, observation O is a conjunction of propositions existentially quantified with the widest possible scope

$$P_1 : c_1 \wedge ... \wedge P_n : c_n \tag{1}$$

where P_i are propositions and c_i are positive real-valued costs ($i \in \{1,..,n\}$). We use the notation $P : c$ to say that proposition P has cost c, and $cost(P)$ to represent the cost of P. The background knowledge B is a set of first-order logic formulas of the form

$$P_1^{w_1} \wedge ... \wedge P_n^{w_n} \rightarrow Q_1 \wedge ... \wedge Q_m \tag{2}$$

where P_i, Q_j are propositions and w_i is a positive real-valued weight ($i \in \{1,..,n\}$, $j \in \{1,..,m\}$). We use the notation P^w to indicate that proposition P has weight w. All variables on the left-hand side of such axioms are universally quantified with the widest possible scope. Variables occurring on the right-hand side only are existentially quantified.[2]

The two main inference operations in weighted abduction are backward chaining and unification. *Backward chaining* is the introduction of new assumptions given an observation and background knowledge. For example, given $O = \exists x(q(x) : 10)$ and $B = \{\forall x(p(x)^{1.2} \rightarrow q(x))\}$, there are two candidate hypotheses: $H_1 = \exists x(q(x) : 10)$ and $H_2 = \exists x(p(x) : 12)$. In weighted abduction, a *cost function* f is used in order to calculate assumption costs. The function takes two arguments: costs of the propositions backchained on and weight of the assumption. Usually, a multiplication function is used, i.e. $f(c,w) = c \cdot w$, where c is the cost of the propositions backchained on and w is the weight of the corresponding assumption. For example, if $q(x)$ costs 10 and w of p is 1.2 in the example above, then assuming p in H_2 costs 12.

Unification is the merging of propositions with the same predicate name by assuming that their arguments are the same and assigning the smallest cost to the result of the unification. For example, $O = \exists x, y(p(x) : 10 \wedge p(y) : 20 \wedge q(y) : 10)$. There is a candidate hypothesis $H = \exists x(p(x) : 10 \wedge q(x) : 10)$. The idea behind such mergings is that if an assumption has already been made then there is no need to make it again.

Both operations (backchaining and unification) can be applied any number of times to generate a possibly infinite set of candidate hypotheses. Weighted abduction defines the *cost* of hypothesis H as

[2] In the rest of this chapter we omit quantification.

$$cost(H) = \sum_{h \in H} cost(h) \tag{3}$$

where h is an atomic conjunct in H (e.g., $p(x)$ in the above H). In this framework, minimum-cost explanations are best explanations. The main idea of weighted abduction is to favor explanations involving fewer assumptions and more reliable assumptions.

3 Discourse Processing Pipeline and Abductive Reasoning

Our discourse processing pipeline produces interpretations of texts given an appropriate knowledge base. A text is first input to the English Slot Grammar (ESG) parser (McCord, 1990, 2010; McCord et al., 2012). For each segment, the parse produced by ESG is a dependency tree that shows both surface and deep structure. The deep structure is exhibited via a word sense predication for each node, with logical arguments. These logical predications form a good start on a logical form (LF) for the whole segment. A component of ESG converts the parse tree into a LF in the style of Hobbs (1985).

The LF is a conjunction of predications, which have generalized entity arguments that can be used for showing relationships among the predications. Hobbs (1985) extends Davidson's approach (Davidson, 1967) to all predications and posits that corresponding to any predication that can be made in natural language, there is an eventuality. Correspondingly, any predication in the logical notation has an extra argument, which refers to the "condition" in which that predication is true. Thus, in the logical form $John(e_1, j) \wedge run(e_2, j)$ for the sentence *John runs*, e_2 is a running event by John and e_1 is a condition of j being named "John".

In terms of weighted abduction, logical forms represent observations, which need to be explained by background knowledge. In the context of discourse processing, we call a hypothesis explaining a logical form an *interpretation* of this LF. In our pipeline, the interpretation of the text is carried out by an inference system called *Mini-TACITUS* (Mulkar-Mehta, 2007). *Mini-TACITUS* tries to prove the logical form of the text, allowing assumptions where necessary. Where the system is able to prove parts of the LF, it is anchoring it in what is already known from the overall discourse or from a knowledge base. Where assumptions are necessary, it is gaining new information. Obviously, there are many possible proofs in this procedure. A cost function on proofs enables the system to chose the "best" (the cheapest) interpretation. The key factors involved in assigning a cost are the following.

1. Proofs with fewer assumptions are favored.
2. Short proofs are favored over long ones.
3. Plausible axioms are favored over less plausible axioms.
4. Proofs are favored that exploit the inherent implicit redundancy in texts.

Let us illustrate the procedure with a simple example. Suppose that we want to construct the best interpretation of the sentence *John composed a sonata*. As

a by-product, the procedure will disambiguate between two readings of *compose*, namely between the "put together" reading instantiated, for example, in the sentence *The party composed a committee*, and the "create art" reading After being processed by the parser, the sentence will be assigned the following logical form, where the numbers (10) after every proposition correspond to the default costs of these propositions.[3] The total cost of this logical form is equal to 30:

$$John(e_1,x_1):10 \wedge compose(e_0,x_1,x_2):10 \wedge sonata(e_2,x_2):10$$

Suppose our knowledge base contains the following axioms:

1) $put\_together(e,x_1,x_2)^{0.6} \wedge collection(e_2,x_2)^{0.6} \rightarrow compose(e,x_1,x_2)$
2) $create\_art(e,x_1,x_2)^{0.6} \wedge work\_of\_art(e_2,x_2)^{0.6} \rightarrow compose(e,x_1,x_2)$
3) $sonata(e,x)^{1.5} \rightarrow work\_of\_art(e,x)$

Axioms (1) and (2) correspond to the two readings of *compose*. Axiom (3) states that a sonata is a work of art. The propositions on the right hand side (*compose*, *work_of_art*) correspond to the given information, whereas the left hand side propositions will be assumed.

Two interpretations can be constructed for the LF above. The first one is the result of the application of Ax. (1). The costs of the backchained propositions (*compose*, *sonata*) are set to 0, because their costs are now carried by the newly introduced assumptions (*put_together*, *collection*). The total cost of the first interpretation **I1** is 32.

I1: $John(e_1,x_1):10 \wedge compose(e,x_1,x_2):0 \wedge sonata(e_2,x_2):10 \wedge$
$put\_together(e_0,x_1,x_2):6 \wedge collection(e_2,x_2):6$

The second interpretation is constructed in several steps. First, Ax. (2) is applied, so that *compose* is backchained on to *create_art* and *work_of_art* with the costs 6. Then, Ax. (3) is applied to *work_of_art*.

I2: $John(e_1,x_1):10 \wedge compose(e,x_1,x_2):0 \wedge sonata(e_2,x_2):10 \wedge$
$create\_art(e_0,x_1,x_2):6 \wedge work\_of\_art(e_2,x_2):0 \wedge sonata(e_2,x_2):9$

The total cost of **I2** is 35. This interpretation is redundant, because it contains the predicate *sonata* twice. The procedure will unify propositions with the same predicate name, setting the corresponding arguments of these propositions to be equal and assigning the minimum of the costs to the result of merging. Thus, the final form of the second interpretation **I2** with the cost of 25 contains only one *sonata* with the cost of 9. The "create art" meaning of *compose* was chosen because it reveals implicit redundancy in the sentence.

[3] The actual value of the default costs of the input propositions does not matter, because the interpretation costs are calculated using a multiplication function. The only heuristic we use here concerns setting all costs of the input propositions to be equal (all propositions cost 10 in the discussed example). This heuristic needs further investigation.

Thus, on each reasoning step the procedure 1) applies axioms to propositions with non-zero costs and 2) merges propositions with the same predicate, assigning the lowest cost to the result of merging. Reasoning terminates when no more axioms can be applied. The procedure favors the cheapest interpretations. Among them, the shortest proofs are favored; i.e. if two interpretations have the same cost then the one that has been constructed with fewer axiom application steps is considered to be "better".

The described procedure provides solutions to a whole range of natural language pragmatics problems, such as resolving ambiguity and discovering implicit relations in noun compounds, prepositional phrases, or discourse structure; see (Hobbs et al., 1993) for detailed examples. Moreover, this account of interpretation solves the problem of where to stop drawing inferences, which could easily be unlimited in number; an inference is appropriate if it is part of the lowest-cost proof of the logical form.

4 Unification in Weighted Abduction

Frequently, the lowest-cost interpretation results from identifying two entities with each other, so that their common properties only need to be proved or assumed once. This feature of the algorithm is called "unification", and is one of the principal methods by which coreference is resolved.

However, this feature of the weighted abduction algorithm has a substantial potential for overmerging. Merging propositions with the same predicate names does not always give the intended solution. If we know $animal(e_1,x)$ and $animal(e_2,y)$, we do not want to assume x equals y if we also know $dog(e_3,x)$ and $cat(e_4,y)$. For *John runs and Bill runs*, with the logical form $John(e_1,x) \wedge run(e_2,x) \wedge Bill(e_3,y) \wedge run(e_4,y)$, we do not want to assume John and Bill are the same individual just because they are both running.

For the full treatment of the overmerging problem, one needs a careful analysis of coreference, including the complicated issue of event coreference. In this study, we adopt a heuristic solution.

The *Mini-TACITUS* system allows us to define non-merge constraints, which prevent undesirable mergings at every reasoning step. Non-merge constraints have the form $x_1 \neq y_1, \ldots, x_n \neq y_n$. These constraints are generated by the system at each reasoning step. Given the propositions $p(x_1)$ and $p(x_2)$ occurring in the input logical form and the non-merge constraint $x_1 \neq x_2$, *Mini-TACITUS* does not merge $p(x_1)$ and $p(x_2)$, because it would imply a conflict with the non-merge constraint. In the experiments described in this book, we used the following rule for generating non-merge constraints.

For each two propositions $p(e_1,x_1,...,x_n)$ and $p(e_2,y_1,...,y_n)$, which occur in the input, if

- e_1 is not equal to e_2,

- p is not a noun predicate, and
- $\exists i \in \{1,\ldots,n\}$ such that x_i is not equal to y_i, and both x_i and y_i occur as arguments of propositions other than $p(e_1,x_1,...,x_n)$ and $p(e_2,y_1,...,y_n)$,

then add $e_1 \neq e_2$ to the non-merge constraints.

This rule ensures that nouns can be merged without any restriction and other predicates can be merged only if all their non-first arguments are equal (due to the previous mergings) or uninstantiated. As seen from the statements above, the argument merging restriction concerns first arguments only. First arguments of all predicates in the logical forms are treated by *Mini-TACITUS* as "handles" referring to conditions, in which the predicate is true of its arguments, i.e. referring to the predication itself, rather than to its semantic arguments.

The proposed non-merge rule is a heuristic, which corresponds to the intuition that it is unlikely that the same noun refers to different entities in a short discourse, while for other predicates this is possible. According to this rule the two *eat* propositions can be merged in the sentence *John eats an apple and he eats the fruit slowly* having the following logical form[4]:

$$John(e_1,x_1) \wedge eat(e_2,x_1,x_2) \wedge apple(e_3,x_2) \wedge and(e_4,e_2,e_5) \wedge$$
$$he(e_1,x_1) \wedge eat(e_5,x_1,x_3) \wedge fruit(e_6,x_3) \wedge slowly(e_7,e_5)$$

In the logical form above, the propositions $eat(e_2,x_1,x_2)$ and $eat(e_5,x_1,x_3)$ cannot be merged, because they do not refer to nouns and their third arguments x_2 and x_3 are not equal. If the knowledge base contains the axiom $apple(e_1,x_1) \rightarrow fruit(e_1,x_1)$ then the logical form above can be expanded into the following:

$$John(e_1,x_1) \wedge eat(e_2,x_1,x_2) \wedge apple(e_3,x_2) \wedge and(e_4,e_2,e_5) \wedge$$
$$he(e_1,x_1) \wedge eat(e_5,x_1,x_3) \wedge fruit(e_6,x_3) \wedge apple(e_6,x_3) \wedge slowly(e_7,e_5)$$

After the expansion, the noun propositions $apple(e_3,x_2)$ and $apple(e_6,x_3)$ can be merged. Now, when all the arguments of the two *eat* propositions are equal, these propositions can be merged as well.

Concerning the sentence *John eats an apple and Bill eats an apple*, merging of two *eat* propositions is impossible, unless the system manages to prove that the predicates *John* and *Bill* can refer to the same individual.

There are cases when the proposed rule does not block undesired mergings. For example, given the sentence *John owns red apples and green apples*, it is wrong to merge both *apple* propositions, because "being red" and "being green" are incompatible properties that cannot be both assigned to the same entity. Thus, it seems to be reasonable to check whether two propositions to be merged have incompatible properties. A detailed study of coreference in an abductive framework is described in (Inoue, 2012).

[4] The anaphoric *he* in the logical form is already linked to its antecedent *John*.

5 Knowledge Base

The proposed discourse processing procedure is based on a knowledge base (KB) consisting of a set of axioms. In order to obtain a reliable KB with a large coverage we exploited existing lexical-semantic resources.

First, we have extracted axioms from WordNet (Fellbaum, 1998), version 3.0., which has already proved itself to be useful in knowledge-intensive NLP applications. The central entity in WordNet (WN) is called a *synset*. Synsets correspond to word senses, so that every lexeme can participate in several synsets. We used the lexeme-synset mapping for generating axioms. For example, in the axioms below, the verb *compose* is mapped to *synset-X*, which represents one of its senses.

$$\textit{synset-X}(s,e) \rightarrow \textit{compose}(e,x_1,x_2)$$

Moreover, we have converted the following WordNet relations defined on synsets into axioms: hypernymy, instantiation, entailment, similarity, and meronymy. Hypernymy and instantiation relations presuppose that the related synsets refer to the same entity (the first axiom below), whereas other types of relations relate synsets referring to different entities (the second axiom below).

$$\textit{synset-1}(e_0,e_1) \rightarrow \textit{synset-2}(e_0,e_1)$$
$$\textit{synset-1}(e_0,e_1) \rightarrow \textit{synset-2}(e_2,e_3)$$

WordNet also provides morphosemantic relations, which relate verbs and nouns, e.g., *buy-buyer*. These relations can be used to generate axioms like the following one.

$$\textit{buyer}(e_1,x_1) \rightarrow \textit{buy}(e_2,x_1,x_2)$$

Additionally, we have exploited the WordNet synset definitions. In WordNet the definitions are given in natural language form. We have used the extended WordNet resource[5], which provides logical forms for the definition in WordNet version 2.0. We have adapted logical forms from extended WordNet to our representation format and converted them into axioms; for example, the following axiom represents the meaning of the synset containing such lexemes as *horseback*.

$$\textit{on}(e_1,e_2,x_1) \wedge \textit{back}(e_3,x_1) \wedge \textit{of}(e_4,x_1,x_2) \wedge \textit{horse}(e_5,x_2) \rightarrow \textit{synset-X}(e_0,x_0)$$

The second resource, which we have used as a source of axioms, is FrameNet, release 1.5, see Ruppenhofer et al. (2006). FrameNet has a shorter history in NLP applications than WordNet, but its potential to improve the quality of question answering (Shen and Lapata, 2007) and recognizing textual entailment (Burchardt et al., 2009) has been demonstrated. The lexical meaning of predicates in FrameNet is represented in terms of frames, which describe prototypical situations spoken about

[5] http://xwn.hlt.utdallas.edu/

in natural language. Every frame contains a set of roles corresponding to the participants of the described situation. Predicates with similar semantics are assigned to the same frame. For example, both *give* and *hand over* refer to the GIVING frame. For most of the lexemes FrameNet provides syntactic patterns showing the surface realization of these lexemes and their arguments. We used the patterns for deriving axioms. For example, the axiom below corresponds to phrases like *John gave a book to Mary*.

$$\text{GIVING}(e_1,x_1,x_2,x_3) \wedge \text{DONOR}(e_1,x_1) \wedge \text{RECIPIENT}(e_1,x_2) \wedge \text{THEME}(e_1,x_3) \\ \rightarrow give(e_1,x_1,x_3) \wedge to(e_2,e_1,x_2)$$

FrameNet also introduces semantic relations defined on frames such as inheritance, causation or precedence; for example, the GIVING and GETTING frames are connected with the causation relation. Roles of the connected frames are also linked, e.g. DONOR in GIVING is linked with SOURCE in GETTING. Frame relations have no formal semantics in FrameNet. In order to generate corresponding axioms, we used the previous work on axiomatizing frame relations and generating new relations from corpora (Ovchinnikova et a., 2010; Ovchinnikova, 2012). An example of an axiomatized relation is given below.

$$\text{GIVING}(e_1,x_1,x_2,x_3) \wedge \text{DONOR}(e_1,x_1) \wedge \text{RECIPIENT}(e_1,x_2) \wedge \text{THEME}(e_1,x_3) \\ \rightarrow \\ \text{GETTING}(e_2,x_2,x_3,x_1) \wedge \text{SOURCE}(e_2,x_1) \wedge \text{RECIPIENT}(e_1,x_2) \wedge \text{THEME}(e_1,x_3)$$

Axiom weights are calculated using the frequency of the corresponding word senses in the annotated corpora. The information about frequency is provided both by WordNet and FrameNet. In our framework, axioms of the type *species* $\rightarrow$ *genus* should have weights greater than 1, which means that assuming *species* costs more than assuming *genus*, because there might be many possible *species* for the same *genus*. The weights of such axioms are heuristically defined as ranging from 1 to 2.

In order to assign a weight w_i to a sense i of a lexeme, we use information about the frequency f_i of the word sense in the annotated corpora. An obvious way of converting the frequency f_i to the weight w_i is the following equation:

$$w_i = 2 - \frac{f_i}{\sum_{1 \leq n \leq |S|} f_n} \tag{4}$$

where S is a set of all senses of the lexeme. All axioms representing relations receive equal weights of 1.2.

Both WordNet and FrameNet are manually created resources, which ensures a relatively high quality of the resulting axioms as well as the possibility of exploiting the linguistic information provided for structuring the axioms. Although manual creation of resources is a very time-consuming task, WordNet and FrameNet, being long-term projects, have an extensive coverage of English vocabulary. The coverage of WordNet is currently larger than that of FrameNet (155 000 vs. 12 000 lexemes).

However, the fact that FrameNet introduces complex argument structures (roles) for frames and provides mappings of these structures makes FrameNet especially valuable for reasoning.

The complete list of axioms we have extracted from these resources is given in table 1. The number of axioms is approximated to the nearest hundred.

Table 1 Statistics for extracted axioms.

Axiom type	Source	Number of axioms
Lexeme-synset mappings	WN 3.0	207,000
Lexeme-synset mappings	WN 2.0	203,100
Synset relations	WN 3.0	141,000
Derivational relations	WN 3.0 (annotated)	35,000
Synset definitions	WN 2.0 (parsed, annotated)	115,400
Lexeme-frame mappings	FN 1.5	49,100
Frame relations	FN 1.5 + corpora	5,300

6 Adapting *Mini-TACITUS* to a Large Knowledge Base

Mini-TACITUS (Mulkar et al., 2007) began as a simple backchaining theorem-prover intended to be a more transparent version of the original *TACITUS* system, which was based on Stickel's *PTTP* system (Stickel, 1988). Originally, *Mini-TACITUS* was not designed for treating large amounts of data. A clear and clean reasoning procedure rather than efficiency was in the focus of its developers. In order to make the system work with the large knowledge base, we had to perform several optimization steps and add a couple of new features.

6.1 Time and Depth Parameters

For avoiding the reasoning complexity problem, we introduced two parameters. A time parameter t is used to restrict the processing time. After the processing time exceeds t the reasoning terminates and the best interpretation so far is output. The time parameter ensures that an interpretation will be always returned by the procedure even if reasoning could not be completed in a reasonable time.

A depth parameter d restricts the depth of the inference chain. Suppose that a proposition p occurring in the input has been backchained on and a proposition p' has been introduced as a result. Then, p' will be backchained on and so on. The number of such iterations cannot exceed d. The depth parameter reduces the number of reasoning steps.

The interaction between the time and depth parameters is shown in Algorithm 1.

Algorithm 1 *Mini-TACITUS* reasoning algorithm: interaction of the time and depth parameters.

Input: a logical form LF of a text fragment, a knowledge base KB,
a depth parameter D, a cost parameter C, a time parameter T
Output: the best interpretation I_{best} of LF

$I_{init} := \{p(e,x_1,\ldots,x_n,C,0) | p(e,x_1,\ldots,x_n) \in LF\}$
$I\_set := \{I_{init}\}$
apply_inference(I_{init})
$Cheapest\_I := \{I | I \in I\_set \text{ and } \forall I' \in I\_set : cost(I) \leq cost(I')\}$
$Best\_I := \{I | I \in Cheapest\_I \text{ and } \forall I' \in Cheapest\_I : proof\_length(I) \leq proof\_length(I')\}$
return I_{best}, which is the first element of $Best\_I$

Subroutine *apply_inference*
Input: interpretation I

while $processing\_time < T$ **do**
  for $\alpha \in KB$ **do**
    for $PropSubset \subseteq I$ such that $\forall p(e,x_1,\ldots,x_n,c,d) \in PropSubset : d < D$ **do**
      if α is applicable to PS **then**
        I_{new} := result of application of α to PS
        $I\_set := I\_set \cup \{I_{new}\}$
        apply_inference(I_{new})
      end if
    end for
  end for
end while

6.2 Filtering out Axioms and Input Propositions

Since *Mini-TACITUS* processing time increases exponentially with the input size (sentence length and number of axioms), making such a large set of axioms work was an additional issue. For speeding up reasoning it was necessary to reduce both the number of the input propositions and the number of axioms. In order to reduce the number of axioms, the axioms that could never lead to any merging are filtered out. Suppose that the initial logical form contains the following propositions:

$$a(x_1,...,x_n) \wedge b(y_1,...y_m) \wedge c(z_1,...,z_k)$$

and the knowledge base consists of the following axioms:

(1) $d(x_1,...,x_l) \rightarrow a(y_1,...,y_n)$
(2) $b(x_1,...,x_m) \rightarrow d(y_1,...,y_l)$
(3) $e(x_1,...,x_t) \rightarrow c(y_1,...y_k)$

Given the logical form above, Ax. (3) is obviously useless. It can be evoked by the input proposition $c(z_1,...,z_k)$ introducing the new predicate e, but it can never lead to any merging reducing the interpretation cost. Thus, there is no need to apply this axiom.

Similarly, proposition $c(z_1,...,z_k)$ in the input logical form can never be merged with any other proposition and can never evoke an axiom introducing a proposition, which can be merged with any other. Therefore, removing the proposition $c(z_1,...,z_k)$ from the input for the reasoning machine and adding it to the best interpretation after the reasoning terminates (replacing its arguments with new variables if mergings took place) does not influence the reasoning process.

In logical forms, propositions that could not be linked to the rest of the discourse often refer to modifiers. For example, consider the sentence *Yesterday, John bought a book, but he has not started reading it yet*. The information concerning John buying a book is in the focus of this text fragment; it is linked to the second part of the sentence. However, the modifier *yesterday* just places the situation in time; it is not connected to any other part of the discourse.

7 Recognizing Textual Entailment

As the reader can see from the previous sections, the discourse processing procedure we have presented is fairly general and not tuned for any particular type of inference. We have evaluated the procedure and the KB derived from WordNet and FrameNet on the recognizing textual entailment (RTE) task, which is a generic task that seems to capture major semantic inference needs across many natural language processing applications. In this task, the system is given a text (T) and a hypothesis (H) and must decide whether the hypothesis is entailed by the text plus commonsense knowledge.

Our approach is to interpret both the text and the hypothesis using *Mini-TACITUS*, and then see whether adding information derived from the text to the knowledge base will reduce the cost of the best abductive proof of the hypothesis as compared to using the original knowledge base only. If the cost reduction exceeds a threshold determined from a training set, then we predict entailment.

A simple example would be the text *John gave a book to Mary* and the hypothesis *Mary got a book*. Our pipeline constructs the following logical forms for these two sentences.

T: $John(e_1,x_1):10 \wedge give(e_0,x_1,x_2):10 \wedge book(e_2,x_2):10 \wedge$
$to(e_4,e_0,x_3):10 \wedge Mary(e_3,x_3):10$
H: $Mary(e_1,x_1):10 \wedge get(e_0,x_1,x_2):10 \wedge book(e_2,x_2):10$

These logical forms constitute the *Mini-TACITUS* input. *Mini-TACITUS* applies the axioms from the knowledge base to the input logical forms in order to reduce the overall cost of the interpretations. Suppose that we have the following FrameNet axioms in our knowledge base.

1) $\textsc{Giving}(e_1,x_1,x_2,x_3)^{0.9} \rightarrow give(e_1,x_1,x_3) \wedge to(e_2,e_1,x_2)$
2) $\textsc{Getting}(e_1,x_1,x_2,x_3)^{0.9} \rightarrow get(e_1,x_1,x_2)$
3) $\textsc{Giving}(e_1,x_1,x_2,x_3)^{1.2} \rightarrow \textsc{Getting}(e_2,x_2,x_3,x_1)$

The first axiom maps *give to* to the GIVING frame, the second one maps *get* to GETTING and the third one relates GIVING and GETTING with the causation relation. As a result of the application of the axioms the following best interpretations will be constructed for T and H.

I(T): $John(e_1,x_1):10 \wedge give(e_0,x_1,x_2):0 \wedge book(e_2,x_2):10 \wedge$
$to(e_2,e_0,x_3):0 \wedge Mary(e_3,x_3):20 \wedge \text{GIVING}(e_0,x_1,x_2,x_3):18$
I(H): $Mary(e_1,x_1):10 \wedge get(e_0,x_1,x_2):0 \wedge book(e_2,x_2):10 \wedge$
$\text{GETTING}(e_0,x_1,x_2):9$

The total cost of the best interpretation for H is equal to 29. Now the best interpretation of T will be added to H with the zero costs (as if T has been totally proven) and we will try to prove H once again. First of all, merging of the propositions with the same names will result in reducing costs of the propositions *Mary* and *book* to 0, because they occur in T:

I(I(T)+H): $John(e_1,x_1):0 \wedge give(e_0,x_1,x_2):0 \wedge book(e_2,x_2):0 \wedge$
$to(e_2,e_0,x_3):0 \wedge Mary(e_3,x_3):20 \wedge \text{GIVING}(e_0,x_1,x_2,x_3):0 \wedge$
$get(e_4,x_3,x_2):0 \wedge \text{GETTING}(e_4,x_3,x_2):9$

The only proposition left to be proved is GETTING. Using the GETTING-GIVING relation in Ax. (3) above, this proposition can be backchained on to GIVING, which will merge with GIVING coming from the T sentence. H appears to be proven completely with respect to T; the total cost of its best interpretation given T is equal to 0. Thus, using knowledge from T helped to reduce the cost of the best interpretation of H from 29 to 0.

In our framework, a full treatment of the logical structure of natural language would require a procedure for assessing the truth claims of a text given its logical form. Quantifiers and logical operators would be treated as predicates, and their principal properties would be expressed in axioms. However, we have not yet implemented this. Without a special account for the logical connectors *if*, *not* and *or*, given a text *If A then B* and a hypothesis *A and B*, our procedure will most likely predict entailment. Even worse, *not A* will entail *A*. Similarly, modality is not handled. Thus, *X said A* and *maybe A* both entail *A*. At the moment our RTE procedure mainly accounts for the informational content of texts, being able to detect the "aboutness" overlap of T and H, and does not reason about the truth or falsity of T and H.

8 Experimental Evaluation

We evaluated our procedure on the RTE-2 Challenge dataset[6] (Bar-Haim et al., 2006). The RTE-2 dataset contains the development and the test set, both including 800 text-hypothesis pairs. Each dataset consists of four subsets, which correspond

[6] http://pascallin.ecs.soton.ac.uk/Challenges/RTE2/

to typical success and failure settings in different applications: information extraction (IE), information retrieval (IR), question answering (QA), and summarization (SUM). In total, 200 pairs were collected for each application in each dataset.

The main task in the RTE-2 challenge was entailment prediction for each pair in the test set. The evaluation criterion for this task was *accuracy* - the percentage of pairs correctly judged. The accuracy achieved by the 23 participating systems ranges from 53% to 75%. Two systems had 73% and 75% accuracy, two systems achieved 62% and 63%, while most of the systems achieved 55%–61% (cf. Bar-Haim et al., 2006).

Garoufi (2007) has performed a detailed study of the RTE-2 dataset investigating factors responsible for entailment in a significant number of text-hypothesis pairs. Surprisingly, Garoufi's conclusion is that such shallow features as lexical overlap (number of words from hypothesis, which also occur in text) seem to be more useful for predicting entailment than any sophisticated linguistic analysis or knowledge-based inference. This fact may have two explanations: Either the RTE-2 dataset is not properly balanced for testing advanced textual entailment technology, or the state-of-the-art RTE systems indeed cannot suggest anything more effective than simple lexical overlap.

Nevertheless, we chose the RTE-2 dataset for our experiments. First, none of the other RTE datasets has been studied in so much detail, therefore there is no guarantee that any other dataset has better properties. Second, the RTE-2 test set was additionally annotated with FrameNet semantic roles, which enables us to use it for evaluation of semantic role labeling.

8.1 Weighted Abduction for Recognizing Textual Entailment

We evaluated our procedure in RTE as described in section 7. The RTE-2 development set was used to train the threshold for discriminating between the "entailment" and "no entailment" cases. Interpretation costs were normalized to the number of propositions in the corresponding H logical forms. This was done in order to normalize over the prediction of longer and shorter hypotheses. If hypothesis h_1 contains more propositions than h_2, then it can potentially contain more propositions not linked to propositions in the text.

As a baseline we processed the datasets with an empty knowledge base. The depth parameter was set to 3. Then, we did different runs, evaluating knowledge extracted from different resources separately.[7] Table 2 contains the results of our experiments.[8] The results suggest that the proposed method seems to be promising as compared to the other systems evaluated on the same task. Our best run gives 62.6% accuracy.

---

[7] The computation was done on a High Performance Cluster (320 2.4 GHz nodes, CentOS 5.0) of the Center for Industrial Mathematics (Bremen, Germany).

[8] "Number of axioms" stands for the average number of axioms applied per sentence.

Table 2 Evaluation results for the RTE-2 test set.

KB	Accuracy	Number of axioms	
		T	H
No KB	57.3%	0	0
WN 3.0	59.6%	294	111
FN	60.1%	1233	510
Ext. WN 2.0	58.1%	215	85
WN 3.0 + FN	62.6%	1527	521

Task	Accuracy
SUM	75%
IR	64%
QA	62%
IE	50%

The obtained baseline of 57.3% is close to the lexical overlap baselines reported by the participants of RTE-2 (Bar-Haim et al., 2006). Although FrameNet has provided fewer axioms than WordNet in total (ca. 50 000 vs. 600 000), its application resulted in better accuracy than application of WordNet. The reason for this might be the confusing fine-grainedness of WordNet, which makes word sense disambiguation difficult. Moreover, the average number of WordNet axioms per sentence is smaller than the number of FrameNet axioms (cf. table 2). This happens because the relational network of FrameNet is much more dense than that of WordNet.

The lower performance of the system using the KB consisting of axioms extracted from extended WordNet (Ext. WN 2.0) can be explained. The axioms extracted from the synset definitions introduce a lot of new lexemes into the logical form, since these axioms define words with the help of other words rather than abstract concepts. These new lexemes trigger more axioms. Finally, too many new lexemes are added to the final best interpretation, which can often be noisy. The WN 3.0 and FN axioms set do not cause this problem, because these axioms operate on frames and synsets rather than on lexemes.

For our best run (WN 3.0 + FN), we present the accuracy data for each application separately (table 2). The distribution of the performance of *Mini-TACITUS* on the four datasets corresponds to the average performance of systems participating in RTE-2 as reported by Garoufi (2007). The most challenging task in RTE-2 appeared to be IE. QA and IR follow, and finally, SUM was titled the "easiest" task, with a performance significantly higher than that of any other task.[9]

Experimenting with the time parameter t restricting processing time (see section 6), we found that the performance of *Mini-TACITUS* increases with increasing time of processing. This is not surprising. The smaller t is, the fewer chances *Mini-TACITUS* has to apply all relevant axioms. Tracing the reasoning process, we found that given a long sentence and a short processing time *Mini-TACITUS* had time to construct only a few interpretations, and the "real" best interpretation was not always among them. For example, if the processing time is restricted to 30 minutes per sentence and the knowledge base contains some hundreds of axioms, then *Mini-TACITUS* has not enough time to apply all axioms up to depth 3 and construct all

[9] In order to get a better understanding of which parts of our KB are useful for computing entailment and for which types of entailment, in future, we are planning to use the detailed annotation of the RTE-2 dataset describing the source of the entailment, which was produced by Garoufi (2007). We would like to thank one of the reviewers of our IWCS 2011 paper which is the basis of this chapter for giving us this idea.

possible interpretations in order to select the best one, while processing a single sentence for 30 minutes is definitely not feasible in a realistic setting. This suggests that optimizing the system computationally could lead to producing significantly better results.

Several remarks should be made concerning our RTE procedure. First, measuring overlap of atomic propositions, as performed by most of the RTE systems (cf. Dagan et al., 2010), does not seem to be the perfect measure for predicting entailment. In the example below, H is fully lexically contained in T. Only one proposition *in* and its arguments pointing to the time of the described event actually make a difference in semantics of T and H and imply "no entailment" prediction.

T: *He became a boxing referee in 1964 and became most well-known for his decision against Mike Tyson, during the Holyfield fight, when Tyson bit Holyfield's ear.*
H: *Mike Tyson bit Holyfield's ear in 1964.*

As mentioned before, a much more elaborate treatment of logical connectors, quantification, and modality is required. In the example below, H is fully contained in T, but there is still no entailment.

T: *Drew Walker, NHS Tayside's public health director, said: "It is important to stress that this is not a confirmed case of rabies."*
H: *A case of rabies is confirmed.*

In order to address some of the problems mentioned above, one can experiment with more sophisticated classification methods (e.g., SVM or Decision Trees). The number of proven/unproven propositions for each part of speech can be used as a specific feature. This solution might reflect the intuition that an unproven verb, preposition, or negation is more likely to imply "no entailment" than an unproven adjective.

Obviously, WordNet and FrameNet alone are not enough to predict entailment. In the example below, our system inferred that *president* is related to *presidential*, Tehran is a part of Iran, *mayor* and *official* can refer to the same person, *runoff* and *election* can mean the same. However, all this information does not help us to predict entailment. We rather need to interpret the genitive *Iran's election* as *Iran holds election* and be able to infer that if there is an election between A and B, then A faces B in the election.

T: *Iran will hold the first runoff presidential election in its history, between President Akbar Hashemi Rafsanjani and Tehran's hard-line mayor, election officials said Saturday.*
H: *Hashemi Rafsanjani will face Tehran's hard-line mayor in Iran's first runoff presidential election ever, officials said Saturday.*

The knowledge needed for RTE has been analysed, for example, in (Clark et al., 2007) and (Garoufi, 2007). In both works, the conclusion is that lexical-semantic relations are just one type of knowledge required. Thus, our knowledge base requires significant extension.

8.2 Semantic Role Labeling

For the run using axioms derived from FrameNet, we have evaluated how well we do in assigning frames and frame roles. For *Mini-TACITUS*, semantic role labeling is a by-product of constructing the best interpretation. But since this task is considered to be important as such in the NLP community, we provide an additional evaluation for it. As a gold standard we have used the Frame-Annotated Corpus for Textual Entailment, FATE (Burchardt and Pennacchiotti, 2008). This corpus provides frame and semantic role label annotations for the RTE-2 challenge test set.[10] It is important to note that FATE annotates only those frames that are relevant for computing entailment. Since *Mini-TACITUS* makes all possible frame assignments for a sentence, we provide only the recall measure for the frame match and leave the precision out.

The FATE corpus was also used as a gold standard for evaluating the *Shalmaneser* system (Erk and Pado, 2006), which is a state-of-the-art system for assigning FrameNet frames and roles. In table 3, we replicate results for *Shalmaneser* alone and *Shalmaneser* boosted with *WordNet Detour to FrameNet* (Burchardt et al., 2005). *WN-FN Detour* extended the frame labels assigned by *Shalmaneser* with the labels related via the FrameNet hierarchy or by the WordNet inheritance relation, cf. Burchardt et al. (2009). In frame matching, the number of frame labels in the gold standard annotation that can also be found in the system annotation (recall) was counted. Role matching was evaluated only on the frames that are correctly annotated by the system. The number of role labels in the gold standard annotation that can also be found in the system annotation (recall) as well as the number of role labels found by the system that also occur in the gold standard (precision), were counted.[11] Table 3 shows that given FrameNet axioms, the performance of *Mini-TACITUS* on semantic role labeling is comparable with those of the system specially designed to solve this task.[12]

---

[10] FATE was annotated with the FrameNet 1.3 labels, while we have been using version 1.5 for extracting axioms. However, in the new FN version the number of frames and roles increases and there is no message about removed frames in the General Release Notes R1.5, see http://framenet.icsi.berkeley.edu. Therefore we suppose that most of the frames and roles used for the FATE annotation are still present in FN 1.5.

[11] We do not compare filler matching, because the FATE syntactic annotation follows different standards as the one produced by the ESG parser, which makes aligning fillers non-trivial.

[12] There exists one more probabilistic system labeling text with FrameNet frames and roles, called *SEMAFOR* (Das et al., 2010). We do not compare our results with the results of *SEMAFOR*, because it has not been evaluated against the FATE corpus yet.

Table 3 Evaluation of frames/roles labeling towards FATE.

System	Frame match	Role match	
	Recall	Precision	Recall
Shalmaneser	0.55	0.54	0.37
Shalmaneser + Detour	0.85	0.52	0.36
Mini-TACITUS	0.65	0.55	0.30

Unfortunately, FrameNet does not really provide any semantic typing for the frame roles. This type of information would be extremely useful for solving the SRL task. For example, consider the phrases *John took a bus* and *the meeting took 2 hours*. The lexeme *take* can be mapped both to the RIDE_VEHICLE and TAKING_TIME frame. Our system can use only the external context for disambiguation of the verb *take*. For example, if the phrase *John took a bus* is accompanied by the phrase *He got off at 10th street*, it is possible to use the relation between RIDE_VEHICLE evoked by *take* and DISEMBARKING evoked by *get off*. However, no information about possible fillers of the roles of the RIDE_VEHICLE frame (living being and vehicle) and the TAKING_TIME frame (activity and time duration) is provided by FrameNet itself. Future work on SRL using FrameNet should include learning semantic preferences for frame roles from corpora.

9 Conclusion and Future Work

This chapter presents a discourse processing framework including the abductive reasoner called *Mini-TACITUS*. We showed that interpreting texts using weighted abduction helps solve pragmatic problems in discourse processing as a by-product. In this chapter, particular attention was paid to reasoning with a large and reliable knowledge base populated with axioms extracted from such lexical-semantic resources as WordNet and FrameNet. The inference procedure as well as the knowledge base were evaluated in the recognizing textual entailment task. The data for evaluation were taken from the RTE-2 Challenge. First, we have evaluated the accuracy of the entailment prediction. Second, we have evaluated frame and role labeling using the Frame-Annotated Corpora for Textual Entailment as the gold standard. In both tasks our system showed performance comparable with those of the state-of-the art systems. Since the inference procedure and the axiom set are general and not tuned for a particular task, we consider the results of our experiments to be promising concerning possible manifold applications of the proposed discourse processing pipeline.

The experiments we have carried out have shown that there is still a lot of room for improving the procedure. First, for successful application of weighted abduction on a large scale the system needs to be computationally optimized. In its current state, *Mini-TACITUS* requires too much time for producing satisfactory results. As our experiments suggest, speeding up reasoning may lead to significant

improvements in the system performance. Since *Mini-TACITUS* was not originally designed for large-scale processing, its implementation is in many aspects not effective enough. Recently, an alternative implementation of weighted abduction based on Integer Linear Programming (ILP) was developed (Inoue and Inui, 2011). In this approach, the abductive reasoning problem is formulated as an ILP optimization problem. In a preliminary experiment the ILP-based system achieved a speed-up over *Mini-TACITUS* of two orders of magnitude (Inoue and Inui, 2011).[13]

Second, in the future we plan to elaborate our treatment of natural language expressions standing for logical connectors such as implication *if*, negation *not*, disjunction *or* and others. Modality and quantifiers such as *all*, *each*, *some* also require a special treatment. This advance is needed in order to achieve more precise entailment inferences, which are at the moment based in our approach on the core information content ("aboutness") of texts. Concerning the heuristic non-merge constraints preventing undesired mergings (see 4), we have performed a detailed study of this issue that is published in Inoue et al. (2012). .

Another future direction concerns the enlargement of the knowledge base. Handcrafted lexical-semantic resources such as WordNet and FrameNet provide both an extensive lexical coverage and a high-value semantic labeling. However, such resources still lack certain features essential for capturing some of the knowledge required for linguistic inferences. First of all, manually created resources are static; updating them with new information is a slow and time-consuming process. By contrast, commonsense knowledge and the lexicon undergo daily updates. This is especially true for proper names. Although some of the proper names have been already included in WordNet, new names appear regularly. In order to accommodate dynamic knowledge, we plan to make use of the distributional properties of words in large corpora. A similar approach is described, for example, in (Peñas and Ovchinnikova, 2012).

Lexical-semantic resources as knowledge sources for reasoning have another shortcoming: They imply too little structure. WordNet and FrameNet enable some argument mappings of related synsets or frames, but they cannot provide a more detailed concept axiomatization. We are engaged in the manual encoding of abstract theories explicating concepts that pervade natural language discourse, such as causality, change of state, and scales, and the manual encoding of axioms linking lexical items to these theories. The core theories should underlie axiomatization of such highly frequent and ambiguous words as *have*. A selection of the core theories can be found at http://www.isi.edu/∼hobbs/csk.html.

We believe that implementation of these improvements and extensions will make the proposed discourse processing pipeline a powerful reasoning system equipped with enough knowledge to solve manifold NLP tasks on a large scale. In our view, the experiments with the axioms extracted from the lexical-semantic resources presented in this chapter show the potential of weighted abduction for natural language processing and open new ways for its application.

---

[13] The discourse processing pipeline including the ILP-based abductive reasoner is available at https://github.com/metaphor-adp/Metaphor-ADP.

References

Bar-Haim, R., I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor (2006). The second PASCAL recognising textual entailment challenge. In *Proc. of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Burchardt, A., K. Erk, and A. Frank (2005). A WordNet Detour to FrameNet. In *Sprachtechnologie, mobile Kommunikation und linguistische Resourcen*, Volume 8.

Burchardt, A. and M. Pennacchiotti (2008). FATE: a FrameNet-Annotated Corpus for Textual Entailment. In *Proc. of LREC'08*, Marrakech, Morocco.

Burchardt, A., M. Pennacchiotti, S. Thater, and M. Pinkal (2009). Assessing the impact of frame semantics on textual entailment. *Natural Language Engineering 15*(4), 527–550.

Clark, P., P. Harrison, J. Thompson, W. Murray, J. Hobbs, and C. Fellbaum (2007). On the Role of Lexical and World Knowledge in RTE3. In *Proc. of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 54–59.

Dagan, I., B. Dolan, B. Magnini, and D. Roth (2010). Recognizing textual entailment: Rational, evaluation and approaches - Erratum. *Natural Language Engineering 16*(1), 105.

Das, D., N. Schneider, D. Chen, and N. A. Smith (2010). SEMAFOR 1.0: A probabilistic frame-semantic parser. Technical Report CMU-LTI-10-001, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Davidson, D. (1967). The Logical Form of Action Sentences. In N. Rescher (Ed.), *The Logic of Decision and Action*, pp. 81–120. University of Pittsburgh Press.

Erk, K. and S. Pado (2006). Shalmaneser - a flexible toolbox for semantic role assignment. In *Proc. of LREC'06*, Genoa, Italy.

Fellbaum, C. (Ed.) (1998). *WordNet: An Electronic Lexical Database* (First ed.). MIT Press.

Garoufi, K. (2007). Towards a Better Understanding of Applied Textual Entailment: Annotation and Evaluation of the RTE-2 Dataset. Master's thesis, Saarland University.

Hobbs, J. R. (1985). Ontological promiscuity. In *Proc. of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, Illinois, pp. 61–69.

Hobbs, J. R., M. Stickel, D. Appelt, and P. Martin (1993). Interpretation as abduction. *Artificial Intelligence 63*, 69–142.

Inoue, N. and K. Inui (2011). ILP-Based Reasoning for Weighted Abduction. In *Proc. of AAAI Workshop on Plan, Activity and Intent Recognition*.

Inoue, N., E. Ovchinnikova, K. Inui and J. R. Hobbs (2012) *Coreference Resolution with ILP-based Weighted Abduction*, Proc. of the 24th International Conference on Computational Linguistics, pp. 1291-1308

McCord, M. C. (1990). Slot grammar: A system for simpler construction of practical natural language grammars. In *Natural Language and Logic: International Scientific Symposium, Lecture Notes in Computer Science*, pp. 118–145. Springer Verlag.

McCord, M. C. (2010). Using Slot Grammar. Technical report, IBM T. J. Watson Research Center. RC 23978Revised.

McCord, M. C., J. W. Murdock, and B. K. Boguraev (2012). Deep parsing in Watson. *IBM J. Res. & Dev. 56*(3/4), 3:1–3:15.

Mulkar, R., J. R. Hobbs, and E. Hovy (2007). Learning from Reading Syntactically Complex Biology Texts. In *Proc. of the 8th International Symposium on Logical Formalizations of Commonsense Reasoning*, Palo Alto, USA.

Mulkar-Mehta, R. (2007). Mini-TACITUS. http://www.rutumulkar.com/tacitus.html.

Ovchinnikova, E. (2012). *Integration of World Knowledge for Natural Language Understanding*. Atlantis Press, Springer.

Ovchinnikova, E., L. Vieu, A. Oltramari, S. Borgo, and T. Alexandrov (2010). Data-Driven and Ontological Analysis of FrameNet for Natural Language Reasoning. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias (Eds.), *Proc. of LREC'10*, Valletta, Malta. European Language Resources Association (ELRA).

Peñas, A. and E. Ovchinnikova (2012). Unsupervised acquisition of axioms to paraphrase noun compounds and genitives. In *Proc. of the International Conference on Intelligent Text Processing and Computational Linguistics*, LNCS, New Delhi, India, pp. 388–401. Springer.

Ruppenhofer, J., M. Ellsworth, M. Petruck, C. Johnson, and J. Scheffczyk (2006). FrameNet II: Extended Theory and Practice. *International Computer Science Institute*.

Shen, D. and M. Lapata (2007). Using Semantic Roles to Improve Question Answering. In *Proc. of EMNLP-CoNLL*, pp. 12–21.

Stickel, M. E. (1988). A prolog technology theorem prover: Implementation by an extended prolog compiler. *Journal of Automated Reasoning 4*(4), 353–380.