Scope Resolution of Logical Connectives in Natural Language Constraints

Shahzad Akbar, Imran Sarwar Bajwa
Dept of Computer Science & IT Islamia University Bahawalpur
Bahawalpur, Pakistan
shahzadakbarbzu@gmail.com
imran.sarwar@iub.edu.pk

ABSTRACT: *Natural Language Constraints have a vital role in Business Organization. The main problem is scope ambiguity when these NL constraints are translated into formal languages. Human beings can understand the context in which these constraints are defined but it is most difficult for a machine to understand the exact meanings of these constraints in their context and this leads to crash the Business System. Therefore before translating these business constraints to OCL, the scope ambiguities should be resolved for correct translation of NL to OCL. For this purpose a new technique is proposed for handling the scope of logical operators used in NL constraints by using the Markov Logic. The subject and scope knowledge and Markov logic integrates into natural language processing. The subject knowledge has been deployed as context knowledge and scope knowledge was acquired from the business constraints. Markov logic was applied to NL constraints for selecting the most possible meaning of an ambiguous NL constraint based on the context. The presented work shows that by handling the identified cases of scope ambiguities of logical operators, we can take correct translation of business constraints into formal specifications.*

Keywords: Natural Language Constraints, Ambiguity, Markov Logic, Context Knowledge, Logical Operators

Received: 18 August 2014, Revised 25 September 2014, Accepted 29 September 2014

1. Introduction

To support their business processes, organizations develop business information systems. These information systems should act in a manner that is consistent with the objectives and policies of the business organization by enforcing business rules. A compact statement about some aspect of a business is called 'Business Rule' which can be expressed in terms that can be directly related to the business, using simple, unambiguous language that is accessible to all interested parties i.e. a business owner, a business analyst, a technical architect and so on. In general, business rules describe how a company conducts its business. [1]

The Business Rules Group (BRG) – defined business rule in both business perspective and information system perspectives:

from the business perspective, a business rule is guidance that there is an obligation concerning conduct, action, practice or procedure within a particular activity or subject, and from the information system perspective, a business rule is a statement that defines or constrains some aspect of the business.[2] Business rules may be captured by business experts, business owners or end users for keeping business works. While the aim of IT professionals to make their applications usable in reality by capturing business rules.

There are three domains regarding to application of BR in IS development process:

Business System (BS), Information System (IS), and Software System(SS).

Business rules should be described in a form relative to business people regarding to BS perspective. The clearest form is natural language. Regarding to IS perspective, this form is ambiguous for specification of business rules.

For this purpose, BRG has provided the specification of Semantics of Business Vocabulary and Business Rules (SBVR) [3]. SBVR is oriented to business people and is designed to be used for business purposes independently of IS designs.
BR must be defined in unambiguous, clear and precise manner to be implemented within SS for later use from the IS perspective. Many different BR modelling methods and techniques are provided for this purpose.

Object-oriented approach is widely adopted and differs from others by its capability to capture the structure (data) and the behaviour (functions) of business system, while others are able to capture only one of them. Recently, the UML is the de-facto standard for object-oriented analysis and design. The OCL as adopted part of the UML, supplements this methodology with possibility to specify system models in more comprehensive and unambiguous way.

Object Constraint Language (OCL) is a formal specification language for describing constraints in the context of UML model. The OCL has been developed as a business modelling language and has its roots in the Syntropy method [4]. It is a part of UML to compliment its modelling because it is not expressive enough to provide all the relevant details of a specification. However, the OCL is side-effects free language, therefore it cannot change anything in the model: the state of the system will never change because of the evaluation of an OCL expression, even though an OCL expression can be used to specify a state change (e.g., in a post-condition) [5].

Furthermore, it is not possible to write program logic or flow control in the OCL as well as express implementation details.

The main characteristics of the OCL are presented as follows according to [5].

1. Both query and constraint language – it is possible to write a query expression of a body of an operation as well as define constraint to some attribute’s value or existence of some object.

2. Mathematical foundation – it is based on mathematical set theory and predicate logic and it has a formal mathematical semantics.

3. Strongly typed language – model elements used in the OCL expressions must conform with types, therefore the OCL expressions can be checked during modeling.

4. Declarative language – modeler can make decisions at a high level of abstraction without going into details how something should be calculated.

5. Object-Oriented analysis and design method.

The main purpose of the OCL could be formulated as follows according to [4]:

Ø Specify invariants on classes, types or stereotypes in class model.

Ø Specify pre and post conditions of operations.

Ø Describe guards on transitions in state diagrams.

Ø Specify target for messages and actions.

Ø Specify derivation rules for any expression over a UML model.

Many researches have been taken in the area of automatic translation of natural language (NL) specifications to formal specifications such as UML (Unified Modelling Language) models [6] and SQL (Structured Query Language) queries [7].

However, the available tools do not fully provide accuracy in real time software development. The key reason of less accuracy is the inherent ambiguity of the natural languages. Mich [8] showed that 71.8% of a sample of NL software specification is ambiguous. Hence, the ambiguous software specifications lead to inconsistent formal specifications of software constraints. In this paper we only discuss the scope ambiguity "A sentence may have more than one sense or context" of the business constraints having logical operators such as Conjunction AND and Disjunction OR. To resolve scope ambiguity, semantic analysis for shallow and deep semantic parsing of English constraints should be conducted. Semantic Analysis relies on the output of syntactic analysis (such as typed dependency [9]) performed by the Stanford parser [10].

Section 2 describes the Problem of scoping logical operators in business constraints specifically conjunction and disjunction. In Section 3 we proposed a framework of new technique for handling the scope ambiguity of logical operators used in NL constraints. Section 4 describes the ambiguity resolution by using Markov logic and methodology of new proposed technique. Conclusion and future work is discussed in the section 5.

2. Description of The Problem

In this section, we discuss the various types of scope ambiguities we came across in translation of English constraints to OCL constraints. Some sentences are ambiguous because we don't know which logical operator determines the meaning of the sentence. Consider the following sentence: "*Ahamd will eat and drink only if his doctor approves*". The sentence has two logical operators:" *and*" and "*only if*". The trouble is that neither is clearly dominant. Is Ahmad going to eat regardless, but drink only if his doctor approves, or does he need his doctor's approval for both activities? In this paper we emphasizes to resolve the scope ambiguity of Conjunction '*AND*' and Disjunction '*OR*'.

When the two clauses (say P, Q) of a sentence are connected by a coordinating conjunction then in addition the truth functionality the AND operator also suggests the following different scopes:

Ø Chronologically sequential to another: "*Ahmad has a balance and withdraws the Cash.*" gives the meaning P and then Q , If P then Q.

Ø The result of another: "*Ali opened the door and went out*." gives the meaning P and therefore Q, P implies Q.

Ø Contrast to another: "*Ali is hard working and Ahmad is conceptual*." gives the meaning P and however Q, P but Q.

Ø Element of surprise: "*Ahmad plays hockey and his favorite sport is cricket.*" gives the meaning P yet Q.

Ø Dependent upon another, conditionally: "*Ali runs fast and reaches the bus stop*." gives the meaning P and thus Q.

In addition to truth functionality the OR operator also suggests the following possible meanings:

Ø Excluding one or the other: "*Are you male or female?*" gives the meaning Either P or Q.

Ø Inclusive combination of alternatives: "*Ali will eat or drink.*" gives the meaning P or Q or Both.

After addressing the identified ambiguity issues related to the scope of conjunction '*AND*' and disjunction '*OR*' sentences, we have presented the framework for handling the scope ambiguity in the next section.

3. Proposed Framework for Handling Scope of Logical Operators

The proposed framework is obtained by considering the use of scope knowledge of logical operators and Markov logic in resolving ambiguity in NL Constraints. The proposed technique applies Scope and subject knowledge as a context and uses Markov logic and possibility distribution to decide the most possible meaning of the logical operators in NL Constraints. The technique is integrated into natural language processing.

The steps in the framework of proposed technique are as follow:

3.1NL Constraint of Logical Operators

All possible business constraints having logical operators are stored in the NL Constraints database.

3.2 Sentence and Subject

A sentence is divided into the Clause 1(P) and Clause 2(Q) in which logical operator is defined. An ambiguous sentence is given as an input. Subject is predefined by a user of the system. The subject which is given as an input along with the sentence is a part of scope knowledge. Knowledge about the scope will be used in calculating the most possible semantic of ambiguous logical constraints which are shown in Table 1 and Table 2 as in below:

Clause 1(P)	***Clause 2(Q)***	***Logical operator***	**Semantics**
P	*Q*	*AND*	*P* have truth functionality to *Q*
P	*Q*	*AND*	*Q* chronologically follows to *P*
P	*Q*	*AND*	*Q* is the result of *P*
P	*Q*	*AND*	*Q* is in contrast to *P*
P	*Q*	*AND*	*Q* is surprise given *P*
P	*Q*	*AND*	*Q* is dependent upon *P*

Table 1. The all Possible Meanings of AND Logical Operator

***Clause 1* (P)**	***Clause 2* (Q)**	***Logical operator***	**Semantics**
P	*Q*	*OR*	*P or Q or Both, Inclusive OR*
P	*Q*	*OR*	*Either P or Q , Exclusive OR*

Table 2. The all possible meanings of OR logical operator

3.3 Scope Knowledge Store

Scope knowledge store is a knowledge-base where it contains a set of NL constraints with their scope. The set of NL constraints with its scope are extracted from the constraint database manually. For instance, NL constraints having OR logical operator may have different possible meanings as shown in Table 3 whereas AND logical operator may have two possible meanings in NL constraints as shown in Table 4. Scope knowledge is generated by combining the input subject and knowledge about scope meanings.

***Clause 1*(P)**	***Clause 2(Q)***	***Logical operator***	**Semantic**	***Scope***
P	*Q*	*OR*	*P* or *Q* or Both, Inclusive *OR*	*P OR Q*
P	*Q*	*OR*	Either *P* or *Q*, Exclusive *OR*	*P XOR Q*

Table 3. Scope Knowledge Database in the Knowledge Store

3.4 Markov Logic Semantic Database

A Markov Logic (ML) semantic database is introduced in this paper. It is produced by utilizing the input subject, scope knowledge, the possibility distribution and Markov logic approach. The context of a sentence is determined by using the subject and scope knowledge database. The ML semantic database is used in resolving ambiguous entities. Details of how ML semantic database is utilized for resolving ambiguous entities will be explained in the next section.

3.5 NLP Techniques

The techniques of NLP that are involved in this work include syntactic processing and semantic processing. Syntactic processing focuses to find out the grammatical structure of the sentence by a grammar and a parser. The output of syntactic processing is a representation of the sentence that reveals the structural dependency relationships between the words. Various grammars that

Clause 1(P)	*Clause 2(Q)*	*Logical operator*	*Semantic*	*Scope*
P	*Q*	*AND*	*P have truth functionality to Q*	*P AND Q*
P	*Q*	*AND*	*Q chronologically follows to*	*P AND then Q*
P	*Q*	*AND*	*P Q is the result of P*	*P AND therefore Q*
P	*Q*	*AND*	*Q is in contrast to P*	*P AND However Q*
P	*Q*	*AND*	*Q is surprise given P*	*P yet Q*
P	*Q*	*AND*	*Q is dependent upon P*	*P AND thus Q*

Table 4. Scope Knowledge Database in the Knowledge Store

can be utilized, and which will, in turn, impact the selection of a parser. All NLP applications do not require a full parse of sentences, therefore the remaining challenges in parsing of prepositional phrase attachment and conjunction scoping are confusing to those applications for which phrasal and clausal dependencies are enough.

Semantic processing determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence. This level of processing can comprises the semantic disambiguation of words with multiple senses. Semantic disambiguation permits only one sense of words to be selected and integrated in the semantic representation of the sentence.

3.6 Unambiguous NL Constraints

Unambiguous sentences that have been identified its semantic based on the context of sentence. As a result of the previous processes, one sentence has only one semantic.

4. Ambiguity Resolution by Markov Logic

In this section, we have presented Markov logic for resolving ambiguous logical operators in details.

4.1 Markov Networks

A Markov network is a model for the joint distribution of a set of variables

$X = (X1 , X2 , . . . , Xn) \in X$ [11]. It consists of an undirected graph G and a set of potential functions φk. The graph has a node for each variable, and the model has a potential function for each clique in the graph. A potential function is a non-negative real-valued function of the state of the corresponding clique. The joint distribution represented by a Markov network is given by

$$P\,(X = x) = \frac{1}{z}\,\Pi_k\,\varphi_k\,(X_{\,(k)})$$

where $x\{k\}$ is the state of the k^{th} clique (i.e., the state of the variables that appear in that clique). Z, known as the partition function, is given by

$$Z = \Sigma_{x \in X}\,\Pi_k\,\varphi_k\,(X_{\,(k)})$$

Markov networks are often represented as log-linear models, with each clique potential replaced by an exponentiated weighted sum of features of the state such as

$$P\,(X = x) = \frac{1}{z}\,exp\left(\Sigma_j\, w_j\, f_j\,(x)\right)$$

A feature may be any real-valued function of the state. there is one feature corresponding to each possible state x{k} of each clique, with its weight being log $\varphi_k\,(x_{\,(k)})$

Markov Logic

A first-order Knowledge base (KB) can be seen as a set of hard constraints on the set of possible worlds: if a world violates even one formula, it has zero probability. The basic idea in Markov logic is to soften these constraints: when a world violates one formula in the KB it is less probable, but not impossible. Each formula has an associated weight that reflects how strong a constraint it is: the higher the weight, the greater the difference in log probability between a world that satisfies the formula and one that does not, other things being equal.

[12] A Markov logic network (MLN) L is a set of pairs (F_i , w_i), where F_i is a formula in first-order logic and w_i is a real number together with a finite set of constants $C = \{c1, c2, . . . , c|C|\}$, it defines a Markov network MLC.

There is an edge between two nodes of MLC if and only if the corresponding ground predicates appear together in at least one grounding of one formula in *L*. An MLN can be viewed as a template for constructing Markov networks. The probability distribution over possible worlds *x* specified by the ground Markov network MLC is given by

$$P\,(X = x) = \frac{1}{z} exp \left(\sum_{i=1}^{F} w_i\, n_i\,(x) \right)$$

where *F* is the number formulas in the MLN and $n_i\,(x)$ is the number of true groundings of F_i in *x*. As formula weights increase, an MLN increasingly resembles a purely logical KB, becoming equivalent to one in the limit of all infinite weights. Now, let us denote *X* as a set of logical operators, *F* denotes a function set of *X* with subject to the context (*C*). Variable *x* is a logical operator may be restricted by the function set *F*. The possibility distribution for *x* to take value under the restriction of *F* is

$$P\,(X = x) = \frac{1}{z} exp \left(\sum_{i=1}^{F} w_i\, n_i\,(x) \right)$$

Now let us consider the scope of the logical operator '*AND*'. The scope context (*C*) to each of the semantics is represented in Table 3 and 4 and is stored in the scope knowledge store. Let us take the '*AND*' as a logical operator *x*, then the scope of *x* can be formalized as $x = m_i\, m_{i+1},..., m_j$ where is the first semantic, and m_j is the last semantic, and its membership function can be derived as $P\,(X = x) = (v_i, v_{i+1},..., v_j)$ where *v* is a plausibility value, and it is context-dependent. When *x* is applied in a different context, it may take a different value. The value of *v* is assigned automatically and randomly by using maximum (max) operator of a function set. Thus $P = (v_i, v_{i+1},..., v_j)$ The most possible semantic can be attached to *x*. In this way, the scope ambiguity can be resolved. ML semantics database is created by applying Markov logic to the technique.

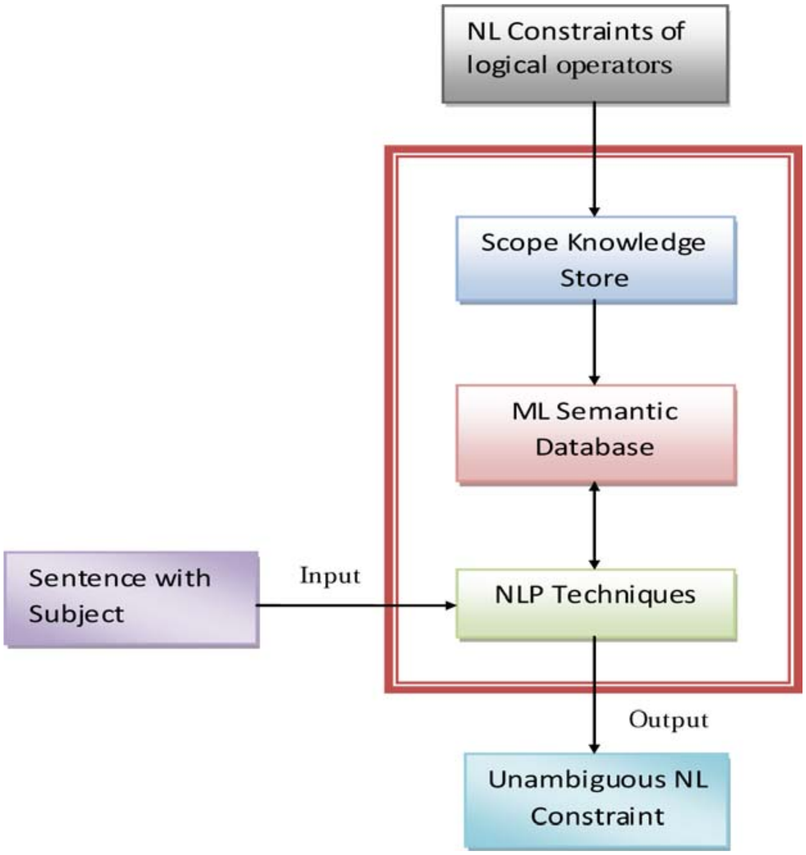

Figure 1. The framework of the proposed technique

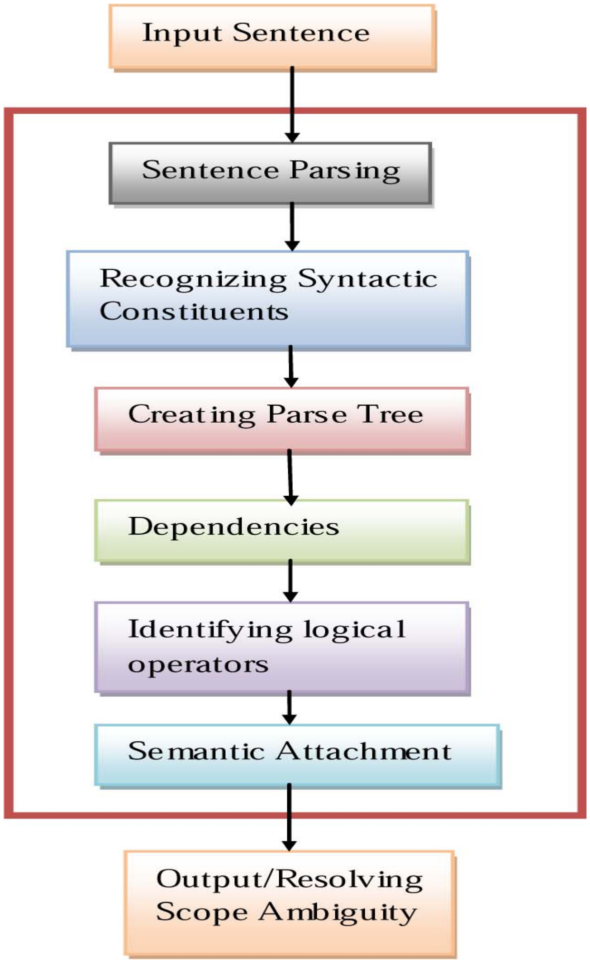
Input Sentence
Sentence Parsing
Recognizing Syntactic Constituents
Creating Parse Tree
Dependencies
Identifying logical operators
Semantic Attachment
Output/Resolving Scope Ambiguity

Figure 2. Methodology of the Proposed Technique

The ML semantic database consists of three fields; logical operator (x), semantic (sem) and semantic value (v). The table can be formalized as

$$T + \{(sem_i , v_i), (sem_{i+1} , v_{i+1}).....(sem_j, v_j))\}$$

where sem_i denotes the scope or semantic of the logical operator x and v is a possibility value attach to it. The value v is in a range of [0, 1] based on the subject context. The values are generated and stored in Table 5 by using human common sense. However, the process of assigning value to a logical operator x is conducted as follow:
Subject: Implication Scope in which conjunction AND gives the meaning of Implication

Table 5 illustrates how values are assigned to semantics randomly. The result of value assignment will be a *ML* semantic database as presented in Table 5. Let us take Implication as a subject example. In the subject of Implication, a grade value of a logical operator '*AND*' to have a semantic of "*P implies Q*" is 0.8 and "*Q is dependent upon P*" is 0.3. If the given subject is "*P yet Q*", its grade value might be 0.8. To resolve an ambiguous logical operator '*AND*', the most possible value *P* is calculated using the max operator of Markov logic. Using the example given in Table 5, the *P* is calculated as:

$P = ((0.5,0.7,0.8,0.4,0.2,0.3))$

The maximum value of all the plausible value in Table 5 is 0.8, which is taken the possible value for the logical operator AND in the context of *P* implies *Q*. The semantic that attach the plausible value 0.8 is "*P implies Q*", therefore, its semantic is taken as the unique semantic of '*AND*'. When the most possible semantic is identified, the semantic attachment is conducted. Once semantic attachment process is completed, semantics for ambiguous logical operators have been resolved. At this stage, all recognized logical operators such as '*AND*' are not ambiguous anymore.

Table 5 illustrates how values are assigned to semantics randomly. The result of value assignment will be a ML semantic database as presented in Table 5. Let us take Implication as a subject example. In the subject of Implication, a grade value of a logical operator '*AND*' to have a semantic of "*P implies Q*" is 0.8 and "*Q is dependent upon P*" is 0.3. If the given subject is "*P yet Q*", its grade value might be 0.8. To resolve an ambiguous logical operator '*AND*', the most possible value *P* is calculated using the max operator of Markov logic. Using the example given in Table 5, the *P* is calculated as:

$P = ((0.5,0.7,0.8,0.4,0.2,0.3))$

The maximum value of all the plausible value in Table 5 is 0.8, which is taken the possible value for the logical operator '*AND*' in the context of P implies Q. The semantic that attach the plausible value 0.8 is "*P implies Q*", therefore, its semantic is taken as the unique semantic of '*AND*'. When the most possible semantic is identified, the semantic attachment is conducted. Once semantic attachment process is completed, semantics for ambiguous logical operators have been resolved. At this stage, all recognized logical operators such as '*AND*' are not ambiguous anymore.

P AND Q (X)	Semantic(S)	Context(C)	ML grade Value
X1	S_X1	C_X1	.5
X2	S_X2	C_X2	.7
X3	S_X3	C_X3	.8
X4	S_X4	C_X4	.4
X5	S_X5	C_X5	.2
X6	S_X6	C_X6	.3

Table 5. ML Semantic Database for "*P implies Q*" CONTEXT

5. Conclusion and Future Work

The main purpose of this paper was how to resolve the scope ambiguity for accurate translation of English constraints to OCL. The results show that after resolving scope ambiguities by using the proposed technique there was significant improvement to correct translation of natural language specifications.

To generalize the presented approach and further improve the accuracy of English to OCL translation we need to work on an automated tool to resolve the scope ambiguity by using the proposed new technique in this paper.

References

[1] Morgan, T. (2002). Business Rules and Information Systems: Aligning IT with Business Goals. Addison Wesley. 2002

[2] Hay, D.C. (2001). et al. Defining Business Rules - What Are They Really?//GUIDE project.

[3] OMG. Semantics of Business Vocabulary and Business Rules, v 1.0. 2008.

[4] OMG. OCL 2.0 specification. 2005.

[5] Warmer, J. (2003). Kleppe A. Object Constraint Language, The: Getting Your Models Ready for MDA, Second Edition. Addison Wesley.

[6] Harmain, H. M., Gaizauskas, R. (2003). CM-Builder: A Natural Language-Based CASE Tool for Object-Oriented Analysis. Automated Software Engineering 10 (2), 157–181.

[7] Giordani, A., Moschitti, A. (2010). Semantic Mapping Between Natural Language Questions and SQL Queries Via Syntactic Pairing. *In*: Horacek, H., Métais, E., Muñoz, R., Wolska, M. (eds.) NLDB 2009. LNCS, 5723, p. 207–221. Springer, Heidelberg (2010).

[8] Mich, L., Franch, M. (2004). Inverardi, P.N.: Market research for requirements analysis using linguistic tools. Requir. Eng., 40–

56 (2004).

[9] Marneffe, M. C., Bill, M., Manning, C. D. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In: LREC.

[10] Cer, D., Marneffe, M. C., Jurafsky, D., Manning, C. D. (2010). Parsing to Stanford Dependencies: Trade-offs between speed and accuracy. *In*: Proceedings of LREC.

[11] Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco, CA.

[12] Richardson, M., Domingos, P. (2006). Markov logic networks. Machine Learning, 62:107–136.