

# The Impact of Semantic Linguistic Features in Relation Extraction: A Logical Relational Learning Approach

Rinaldo Lima<sup>1</sup>, Bernard Espinasse<sup>2</sup>, Fred Freitas<sup>3</sup>

<sup>1</sup> Departamento de Computação, Universidade Federal Rural de Pernambuco, Brazil

<sup>2</sup> LIS UMR CNRS 7020, Aix-Marseille Université/Université de Toulon, France

<sup>3</sup> Centro de Informática, Universidade Federal de Pernambuco, Brazil

rinaldo.jose@ufrpe.br, bernard.espinasse@lis-lab.fr, fred@cin.ufpe.br

## Abstract

Relation Extraction (RE) consists in detecting and classifying semantic relations between entities in a sentence. The vast majority of the state-of-the-art RE systems relies on morphosyntactic features and supervised machine learning algorithms. This paper tries to answer important questions concerning both the impact of semantic-based features, and the integration of external linguistic knowledge resources on RE performance. For that, a RE system based on a logical and relational learning algorithm was used and evaluated on three reference datasets from two distinct domains. The yielded results confirm that the classifiers induced using the proposed richer feature set outperformed the classifiers built with morphosyntactic features in average 4% (F1-measure).

## 1 Introduction

Relation Extraction (RE) consists in detecting and classifying binary semantic relations between entities in a sentence.

Many RE systems use statistical machine learning techniques, such as feature-based and tree kernels-based methods (Choi et al., 2013) are based on a propositional hypothesis space for representing examples, i.e., they employ an attribute-value representation achieving robust results. However, this representation is not able to effectively capture structural information from parse trees without loss of information (Choi et al., 2013). More recently, other RE systems using deep learning techniques, such as Convolution Neural Networks (Nguyen and Grishman, 2015) and Recurrent Neural Networks (Miwa and Bansal, 2016) have also been

proposed. They are based on a dense vector representation of the input words retrieved from word embeddings (Mikolov et al., 2013).

Some other RE systems rely on natural language processing (NLP) techniques for extracting relevant features from the text. They typically integrate shallow NLP tools for coping with lexical and syntactic aspects of the texts such as POS tagging, lemmatization, chunking, and syntactic parsing (Choi et al., 2013). However, according to Zouaq (2011), there are two main reasons to seriously consider deeper linguistic processing for RE: (i) it may provide deeper semantic meaning; (ii) NLP tools are becoming sufficiently robust to be considered as reliable tools for knowledge and model extraction.

Contrarily to related work above, this work subscribes to the idea of performing RE employing the Logical and Relational Learning (LRL) (de Raedt, 2008) approach that can generate classification models from complex data structures such as graphs or multiple tables. More precisely, we rely on Inductive Logic Programming (ILP) (Muggleton, 1991) as one of the most successful relational learning techniques because it employs a symbolic and declarative representation for the examples and the extraction models are both understandable and interpretable by humans. Moreover, ILP allows for many forms of prior knowledge, including semantic resources, to be integrated into the induction of the extraction rules (de Raedt, 2018).

In this paper, we try to answer experimental questions concerning not only the impact of semantic linguistic features but also the integration of external knowledge resources on RE. For that, an ILP-based RE system was employed and evaluated on three datasets from two distinct domains.

The main contribution of this work consists in the experimental validation of our working hypothesis that a feature engineering step comprising a sub-

stantial body of deep linguistic knowledge, in combination with an expressive inductive learning technique, can generate effective RE models.

## 2 Relational Learning and Inductive Logic Programming

Relational Learning (RL) concerns the learning task from complex, heterogeneous examples represented by multirelational datasets. RL enables the development of applications in many fields including bio-informatics, networks analysis, and drug design (de Raedt, 2008). LRL (DeRead, 2008) combines machine learning and logic-based formalisms to automatically induce first-order rules from multi-relational examples.

ILP (Muggleton, 1991) is one of the most successful LRL-based technique that can not only induce symbolic rules from examples represented as multi-relational data, but also integrate background knowledge (BK) represented as logical clauses in first-order logic (FOL). Unlike traditional machine learning methods, classification models (rules) in ILP are both understandable and interpretable by humans. Most of the current ILP systems induce a set of Horn-clauses and employ Prolog as their core inference engine (Muggleton, 1991).

## 3 Logical Relational Learning System for Relation Extraction

The main contribution of this paper consists in the experimental validation of our working hypothesis that a feature engineering step composed by a substantial body of deep linguistic knowledge in combination with an expressive inductive learning technique can generate effective RE models. For testing this hypothesis, a LRL RE system (Lima et al., 2017; Lima et al., 2019) was used. The remainder of this section describes the RE system, the rich feature engineering component, and the underlying model for representing semantic features.

### 3.1 System Architecture

Fig. 1 shows the functional architecture our LRL system for RE. Its major components are highlighted as darker boxes and briefly presented next.

**Deep NLP Component.** This component performs the automatic annotation of the input documents in English. Its output is formed by XML files containing several layers of linguistics annotations. Its distinguishing characteristic consists in various NLP analysis it performs, starting from

tokenization, passing for shallow analysis, and finishing with more advanced semantic analysis.

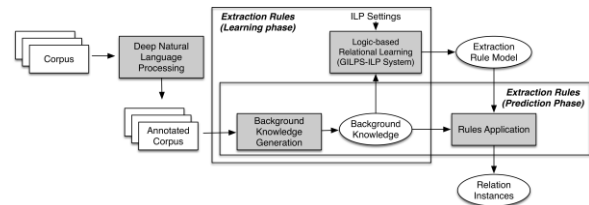


Figure 1. RE System architecture.

**Background Knowledge Generation component.** This component automatically generates and represents relevant features from an annotated set of documents. The generated features are converted into a knowledge base implemented as a Prolog factual base.

**ILP rule learning component** relies on an LRL system rooted on ILP that induces Horn-like extraction rules from training data. The extraction rules follow the same syntax of a Prolog predicate. This learning component is based on GILPS, a general ILP system proposed by Santos (2010).

**Rule application component.** It applies the induced rules on the Prolog factual base generated from new documents not seen in the rule model learning phase. As a result, new instances of relations are identified and extracted.

Due to their importance, the above first two components will be detailed next.

### 3.2 Feature Engineering via Deep NLP

NLP technologies are of paramount importance in RE, since they can analyze unstructured texts and extracting their meaning. Indeed, the result of NLP analyses (or *annotation process*) is a richer version of the input text with further lexical, syntactical, and semantic metadata seem as a normalized representation of texts.

Due to its inherent complexity, NLP is not carried out in a single large stage. Instead, the annotation process is carried out as a chain of processes in which the output of a previous process becomes the input to the next one. Accordingly, the NLP component in our RE architecture is composed of the three major components as depicted in Fig. 2. The first analysis provides the basic morphological elements and lemmas, i.e., canonical base form of the words. In addition, categorical information is attached to each lexical item as Part-of-Speech (POS) tags (e.g., noun, verb, etc.). This facilitates the posterior task of determining groups of words (chunking analysis) that grammatically belong to the same category. Next, the syntactical

analysis (syntactic parsing) identifies the structural relationships holding between words at the sentence level. The final semantic analysis links words to lexical semantic resources, including WordNet (Fellbaum, 1998), SUMO ontology

(Niles and Pease, 2003), and WordNet Domain hierarchy (Bentivoli et al., 2004). Such semantic resources offer a variety of semantic relations including synonyms and hyponyms from WordNet, and additional semantic relations between verbs and their arguments considered here as predicates.

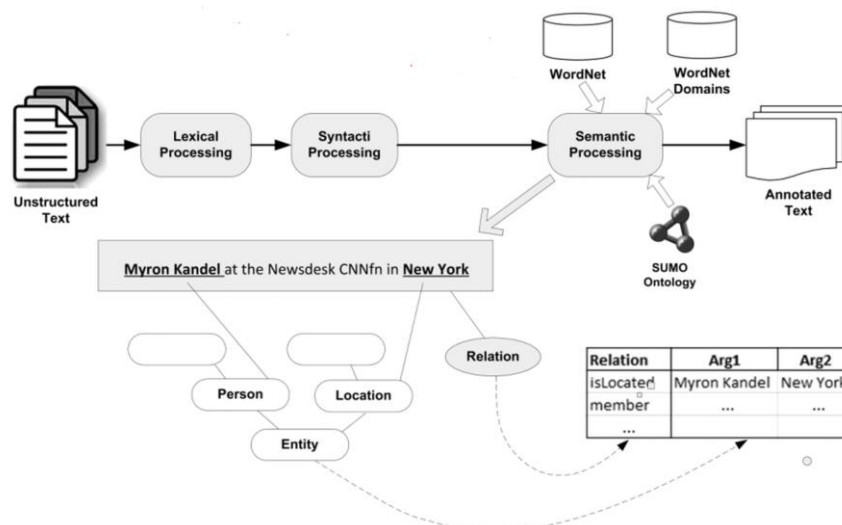


Figure 2. Overview of the Deep NLP pipeline.

The Deep NLP component relies on the Stanford CoreNLP<sup>1</sup> for carrying out the following pipeline: tokenization, sentence splitting, POS tagging, lemmatization, NER, and dependency parsing. Chunking analysis is performed by Apache OpenNLP<sup>2</sup>, while morphological analysis, gazetteers look-up, and pronoun normalization were implemented as ad hoc programs. That is followed by the Sense Learner (Mihalcea and Faruque, 2004) that disambiguates noun and verbs. Then, using the Java WordNet library<sup>3</sup>, the *sense\_id* of nouns and verbs are found in WordNet, along with of all synonyms and hyponyms of a given word. In addition, Lin’s similar words dataset (Lin et al., 2003) is used for retrieving a list of  $N$  ( $N = 5$ ) most similar words to a given word. Semantic Role Labeling (SRL) is performed on all verbs by ClearNLP<sup>4</sup>. Still, for verbs, the SuperSense Tagger<sup>5</sup> (Ciaramita and Altun, 2006) finds their selectional preferences. It annotates text with 41 broad semantic categories (WordNet supersenses) for both nouns and verbs. Next, the mapping between all WordNet synsets and the SUMO ontology<sup>6</sup> is exploited for retrieving the related class from the SUMO ontology. Finally, an ad-

hoc program maps words to labels from the WordNet Domains (Bentivogli *et al.*, 2004). Table 1 summarizes the entire pipeline.

### 3.3 Relational Representation of Sentences

The goal of the BK Generation Component is to extract and represent features according to a relational model consisting of documents, sentences, phrases, and tokens. They are converted into a Prolog factual base and used as input by the ILP-based learner. There are the four main groups of features:

*Lexical features* which concern word, lemma, length, and general morphological type information at the token level.

*Syntactic features* denote word POS tags; head word of nominal, prepositional or verbal chunk; bi-grams and tri-grams of consecutive POS tags of words, chunking features that segment sentences into a noun, prepositional, and verb phrases, chunk head word, and its relative position to the main verb of the sentence.

*Semantic features* include named entities, entity mentions provided by the corpus as well as all lexical-based semantic features (sense/hypersense,

<sup>1</sup> <https://stanfordnlp.github.io/CoreNLP>

<sup>2</sup> <http://opennlp.apache.org>

<sup>3</sup> <https://sourceforge.net/projects/jwordnet>

<sup>4</sup> <https://github.com/clearnlp/clearnlp>

<sup>5</sup> <https://sourceforge.net/projects/supersensetagger>

<sup>6</sup> <http://www.adampease.org/OP>

synonyms, domain sense), and semantic roles of the verbs with their arguments.

*Structural features* consist of the structural elements connecting all the above features according to our relational model. They denote (i) the sequencing of tokens preserving the token order in the input sentence; (ii) the part-whole relation between tokens and the chunk containing them; (iii) the sequencing of chunks is represented by edges between their head tokens; and (iv) the grammatical dependency between two tokens in a sentence according to the typed dependencies between words.

The relational representation of all the above types of features is straightforward: a unary predicate in Prolog denotes identifiers, e.g., *token(id)*, while binary predicates correspond to attribute-value pairs and relations, e.g., *rel(arg1, arg2)*.

NLP Subtask	Tool or Resource
Tokenization	Stanford CoreNLP
Sentence Splitter	
POS	
Lemmatization	
Chunking	OpenNLP Chunker
NER	Stanford CoreNLP
Morphological Analysis	ad hoc programs
Gazetteer Look-up	
Pronoun Normalization	
Syntactic Parsing - Dependency	Stanford CoreNLP
Word Sense Disambiguation	Sense Learner
WordNet Synsets (synonyms and hypernyms)	WordNet 3.0
Similar words	Lin's database
SRL with Propbank/VerbNet	ClearNLP
Selectional Preferences	SuperSense Tagger
Semantic mapping to Domains	WordNet domains
Semantic mapping to SUMO	Ad hoc program

Table 1. Complete pipeline of the Deep NLP tools component.

## 4 Experiments

This section reports the results of experiments performed on three benchmark datasets from two distinct domains (newswire and biomedical).

### 4.1 Experimental Questions

We investigate the effectiveness of the proposed semantic linguistic features used in the induction of the relation extraction rules by the our ILP system. More precisely, we want to answer the following experimental questions (EQ):

**EQ1.** *Do the features present a complementary contribution to the performance results?*

**EQ2.** *What is the impact of the semantic linguistic features on the final induced set of extraction rules?*

**EQ3.** *How well do the rules generalize among different datasets: either in the same domain or in distinct domains?*

### 4.2 Dataset and Evaluation Measures

Three publicly available RE datasets from the newswire and the biomedical domain containing binary relations were selected for analysis:

**reACE 2004/2005.** The reACE 2004/2005 datasets introduced by Hachey et al. (2011) are the result of several transformation steps (refactoring, preprocessing, and reannotation) for normalizing the two original ACE datasets so that they adhere to a common notion of relation that is more intuitive and simpler: relation instance denotes a predicate over two arguments, where the arguments represent concepts in the real world.

Table 2 shows the distributions of the relation types in reACE 2005 dataset, whereas Table 3 shows some examples of them.

reACE 2005 - Relation Types	Freq
Employment	228
Membership	36
Located	280
Citizen-Resident-Religion-Ethnic	39
Business	16
Family	42
Geographical	119
Subsidiary	47

Table 2: reACE 2005 relation types.

Relation type	Example phrases
business (John, superiors)	John's superiors...
Employ (Investors, "Wall Street")	Investors on Wall Street...
citizen (voters, Missouri)	Some Missouri voters...

Table 3. Examples of reACE 2005 relations.

**IEPA.** The Interaction Extraction Performance Assessment (IEPA) corpus (Ding et al., 2002) is a biomedical dataset comprising 303 abstracts retrieved by ten queries suggested by domain experts to the PUBMED repository. An interaction between two terms, i.e., a specific pair of co-occurring chemicals in the IEPA corpus, was defined as a direct or indirect influence of one on the quantity or activity of the other (Ding et al., 2002). Examples of interactions between terms A and B are "*A increased B*", and "*A activated C, and C activated B*".

**Evaluation Measures.** The classical IR measures of Precision *P*, Recall *R*, and F1-measure (Baeza-Yates and Ribeiro-Neto, 1999) were used for measuring the effectiveness (impact) of the proposed enhanced features on the RE task.

### 4.3 Experimental Protocol

We employed five-fold cross-validation which allows both the maximal use of the available training data on all the datasets used in the experiments. Moreover, preliminary experiments were performed for determining the optimal learning parameters according to the criteria of achieving high accuracy and preventing model overfitting. The best parameter setting for the ILP-based learning component found were:  $evalfn = coverage$ ,  $i(depth) = 3$ ,  $minpos = 5$ , and  $noise = 0.2$ .

### 4.4 Results

Table 4 summarizes the results of using several combinations of features on reACE 2004/2005 and IEPA datasets, while Tab. 5 conveniently displays the difference in performance between each pair of corresponding lines indexed by the column id.

Starting from Line 2 in Tab. 4, a given group of features are incrementally added to the baseline (Line 1) which, in turn, includes the following group of features: lexical, syntactic and structural features, i.e., syntactical dependencies (Dep), chunk information (Chunk), POS tagging, and other chunk related features. The baseline setting corresponds to all the features that do not take into account the semantic features (i.e., lexical semantics and mapping to semantic resources). The other lines (Line 2-4) in Tab. 4 integrate other groups of features (NER, Corpus types) to the baseline: NER denotes recognized named entities whereas Corpus types features denote the golden standard annotations already available in the given corpus. For instance, the reACE datasets provide named entities such as *Organizations* and *Person*, while the IEPA corpus only assigns the label *protein* to each term denoting a given protein. The last group of features (*semantic*), denotes the semantic features comprising SRL, synonyms/hypernyms, and mapping of words to WordNet, WordNet Domains, SUMO ontology, similar words, and selectional preferences. The missing entries in IEPA column are due to the fact that typical named entities are useless in the IEPA biomedical corpus, and therefore, they were not considered.

## 5 Discussion on Experimental Questions

This section discusses both the impact of semantic linguistic features on RE, and related aspects on domain adaptability.

### On the Impact of Semantic Linguistic Features.

EQ1 can be positively answered because by incrementally incorporating new groups of features to the baseline, that contributed to the improvement of the scores for all datasets. Indeed, the performance improves as more features are used, starting with the F-measure of 77.77 and reaching 81.80 for the reACE 2004 dataset. Analogously for the reACE 2005, the best overall F1 performance (71.86%) may indicate that this dataset is more difficult than the reACE 2004. One possible explanation is that, in the reACE 2005 dataset, some relations (particularly *Business*) are very poorly represented with only 16 positive examples, which hampers the overall score. More importantly, the overall F1 scores suggest that the proposed four groups of features have both a positive and complementary impact on the overall F1 scores for all the datasets evaluated.

Concerning EQ2, one can notice that including semantic features into the RE process improves average performance in terms of F1 measure for all datasets. In fact, the boost in F1 measure was 4% in average for the reACE datasets, while for the IEPA dataset, the improvement was more than 3%. However, the impact on both P and R scores were unbalanced for the reACE corpora, since the semantic features contributed relatively more in recall than in precision. This contrast with the results on the IEPA corpus that were very balanced. On the one hand, the highest difference in performance was achieved on the reACE 2005 corpus, as the semantic features improved P in almost 12%. On the other hand, for two other combinations in this dataset (Line 5 and Line 7), adding semantic features to the learner in fact slightly hampered precision. Such impact on both P and R were expected since the effect of adding semantic features to the learner could not only improve R over P, but also provide to it an extended layer of categorization of all terms. Contrastingly, for the IEPA corpus, the use of semantic features slightly increased precision more than recall. After inspecting the final induced extraction rules, we found that this was mainly due to the semantic role labeling features. Actually, many verbs denoting the interaction between two proteins terms in IEPA corpus were correctly annotated along with the roles of its arguments. As a result, the ILP-based learner is more precise when a given verb has semantic role features attached to it.

ID	Combination of features	reACE 2004			reACE 2005			IEPA		
		P	R	F1	P	R	F1	P	R	F1
1	Baseline	90.68	66.68	77.77	83.68	50.43	62.91	60.22	72.33	65.72
2	Baseline + <i>NER</i>	92.32	67.12	77.99	80.59	51.39	62.68	-	-	-
3	Baseline + <i>Corpus types</i>	90.30	71.13	80.31	83.03	63.38	71.86	64.40	75.21	69.39
4	Baseline + <i>Corpus types</i> + <i>NER</i>	92.23	73.07	81.80	82.30	61.85	70.62	-	-	-
5	Baseline + <b>Semantic</b>	93.30	70.68	79.44	82.86	59.71	69.41	62.83	74.70	68.25
6	Baseline + <i>NER</i> + <b>Semantic</b>	93.04	75.49	83.06	83.30	60.95	70.40	-	-	-
7	Baseline + <i>Corpus types</i> + <b>Semantic</b>	92.20	77.53	83.43	82.29	63.90	71.94	68.91	79.35	73.76
8	Baseline + <i>Corpus types</i> + <i>NER</i> + <b>Sem.</b>	92.91	79.50	85.39	94.24	62.99	75.10	-	-	-

Baseline = Dep + Chunk + POS + Chunk related

NER = Named entity recognition

Corpus types = golden standard labels of the corpus (reACE and IEPA)

Semantic = SRL, WN synonyms/hyponyms (baseline sense), mapping to WN Domains and SUMO ontology, normalization, similar words, selectional preferences.

Table 4. Results on reACE 2004/2005 and IEPA datasets.

Difference	reACE 2004			reACE 2005			IEPA		
	$\Delta P$	$\Delta R$	$\Delta F$	$\Delta P$	$\Delta R$	$\Delta F$	$\Delta P$	$\Delta R$	$\Delta F$
Line 5 - Line 1	2.62	4.00	1.67	-0.82	9.28	6.50	2.60	2.40	2.54
Line 6 - Line 2	0.72	8.37	5.07	2.71	9.56	7.72	-	-	-
Line 7 - Line 3	1.90	6.40	3.12	-0.74	0.52	0.08	4.50	4.10	4.35
Line 8 - Line 4	0.68	6.43	3.59	11.94	1.14	4.48	-	-	-
Average	1.48	<b>6.30</b>	3.36	3.27	5.13	<b>4.69</b>	<b>3.55</b>	3.25	3.45

Table 5. Performance difference between RE models

**On Domain Adaptability.** Some authors investigated the domain adaptation problem on NER.

Ciaramita et al. (2005) studied the effects of domain adaptation on NER using two distinct datasets for training and testing. They trained several NER classifiers on the CONLL 2003 dataset and evaluated them on a manually annotated section from the Wall Street Journal portion of the Penn Treebank. They found that, even for such similar types of texts, the results of their supervised NER models dropped significantly. However, they had improved scores (almost 5% in F1-measure) just by coupling their original NER system with both a domain-independent dictionary and a simple string similarity function.

In (Pyysalo, 2008), very similar results were reported when a general POS tagger was employed for tagging a dataset from the biomedical domain.

In our work, this same trend was observed since our RE models trained with all the linguistic features yielded, in average, a relative gain up to 4% in F1-measure. In fact, our RE models trained with deep semantic features outperformed the RE models that did not use them according to the statistical significant tests (Wilcoxon signed-rank) for the difference between F1 scores at  $\alpha = 0.05$  (95% confidence interval). These results are very encouraging and seem to validate the proposed features even

in a more difficult application scenario handling the changing of domains. To conclude, the overall achieved results suggest that more accurate semantic information about entity instances can contribute a great deal to RE. This is not surprising, given that semantic information, e.g., hypernyms, and classes from an ontology, typically impose strong constraints on the types of the entities participating in a relation, indicating that such kind of feature can have significant discriminative power in RE. Thus, we can also positively answer to EQ3.

## 6 Conclusion and Future Work

This paper presented a LRL system for RE employed to test of our hypothesis that a set of features based on a deep linguistic analysis can improve RE. This was demonstrated by experimental evaluation showing that automatic acquisition of a substantial body of linguistic knowledge in combination with an expressive inductive learning technique, it is possible to generate effective RE models. Moreover, such features significantly contributed to generalize the proposed RE to other domains of interest.

This research work can be improved in several ways. We intend to test our solution on larger datasets aiming to promote future scalability. In addition, mechanisms for allowing the parallel execution of the IE process will enable the decomposition of the learning problem into smaller more manageable ones. Finally, the system will be adapted Event Extraction, a fundamental IE task in the biomedical domain (Björne and Salakoski, 2015).

## References

- Amal Zouap. 2011. An Overview of Shallow and Deep Natural Language Processing for Ontology Learning. In *W. Wong, W. Liu, & M. Bennamoun (Eds.), Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances*. Hershey, PA: IGI Global, pages 16-37.
- Baeza-Yates, R., Ribeiro-Neto, B., 1999. *Modern Information Retrieval*. Addison-Wesley, Boston.
- Bentivogli L., Forner P., Magnini B. and Pianta E. 2004. Revising WordNet Domains Hierarchy: Semantics, Coverage, and Balancing. In *COLING Workshop on Multilingual Linguistic Resources*, Geneva, Switzerland, pp. 101-108.
- Björne, J., Salakoski, T., 2015. TEES 22: Biomedical event extraction for diverse corpora. *BMC Bioinformatics* 16 (Suppl 16), S4, PMC Web.
- Choi S.P, S. Lee., H. Jung, and S. Song. 2013. An Intensive Case Study on Kernel-based Relation Extraction. In *Proceedings of Multimedia Tools and Applications, Springer, US, 2013, pp. 1 -27*.
- Ciaramita M., Y. Altun. 2005. Named-entity recognition in novel domains with external lexical knowledge. *Adv. in Structured Learning for Text and Speech Processing Workshop (NIPS)*.
- Ciaramita M., Yasemin Altun. 2006. Broad-Coverage Sense Disambiguation and Information Extraction with a Supersense Sequence Tagger. *EMNLP*, pages 594- 602.
- Ding J., Berleant, D., Nettleton, D., and Wurtele, E. 2002. Mining MEDLINE: abstracts, sentences, or phrases? In *Proc. of the Pacific Symposium on Bio-computing*, 326-337.
- Fellbaum, C.D., 1998. *WordNet – An Electronic Lexical Database*. Language, Speech and Communication. MIT Press.
- Hachey H., C. Grover, and R. Tobin. 2011. Datasets for Generic Relation Extraction. *Journal of Natural Language Engineering*, Cambridge University Press.
- Lin D., Zhao S., Qin L., Zhou M. 2003. Identifying Synonyms among Distributionally Similar Words. *IJCAI 2003*: pages 1492-1493.
- Lima R., B. Espinasse, F. Freitas. 2017. OntoILPER: an Ontology- and Inductive Logic Programming-based Relations from Text, in: *Knowledge And Information System (KAIS) Journal*, Springer London.
- Lima, R., Espinasse, B. Freitas, F. 2019. A logic-based relational learning approach to relation extraction: The OntoILPER system. *Journal of Engineering Applications of Artificial Intelligence*. Volume 78, pages 142-157.
- Mihalcea R. and E. Faruque. 2004. SenseLearner: Minimally supervised word sense disambiguation for all words in open text”. In *Proceedings of ACL/SIG-LEX Senseval-3*, Barcelona, Spain.
- Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at ICLR*.
- Miwa, M., & Bansal, M. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, pages 1105–1116.
- Muggleton S. 1991. “Inductive Logic Programming” *New Generation Computing* 8 (4): 29.
- Nguyen, T.H., Grishman, R. 2015. Relation extraction: Perspective from convolutional neural networks. In: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. pages 39–48.
- Niles, I. and Pease, A. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. 2003. In *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE 03)*, Las Vegas, Nevada, June 23-26.
- Pyysalo S. 2008. A dependency parsing approach to biomedical text mining, Ph.D. thesis. *Department of Information Technology, University of Turku*, 2008.
- Luc De Raedt. 2008. Logical and Relational Learning: From ILP to MRDM (*Cognitive Technologies*). Springer-Verlag, Berlin, Heidelberg.
- Santos J. 2010. Efficient Learning and Evaluation of Complex Concepts in Inductive Logic Programming. Ph.D. Thesis, *Imperial College*.