A Formal Approach to Integrating Logical Representations with Probabilistic Information using Markov Logic [TODO: ?]

Dan Garrette, Katrin Erk, and Raymond Mooney

Abstract First-order logic provides a powerful and flexible mechanism for representing natural language semantics. However, it is an open question of how best to integrate it with uncertain, probabilistic knowledge, for example regarding word meaning. This paper describes the first steps of an approach to recasting first-order semantics into the probabilistic models that are part of Statistical Relational AI. Specifically, we show how Discourse Representation Structures can be combined with distributional models for word meaning inside a Markov Logic Network and used to successfully perform inferences that take advantage of logical concepts such as factivity as well as probabilistic information on word meaning in context.

1 Introduction

[TODO: Example numbering should be (1) instead of “Example 1.”]

Logic-based representations of natural language meaning have a long history. Representing the meaning of language in a first-order logical form is appealing because it provides a powerful and flexible way to express even complex propositions. However, systems built solely using first-order logical forms tend to be very brittle as they have no way of integrating uncertain knowledge. They, therefore, tend to have high precision at the cost of low recall (Bos and Markert, 2005).

Dan Garrette
University of Texas at Austin, e-mail: `dhg@cs.utexas.edu`

Katrin Erk
University of Texas at Austin e-mail: `katrin.erk@mail.utexas.edu`

Raymond Mooney
University of Texas at Austin e-mail: `mooney@cs.utexas.edu`

Recent advances in computational linguistics have yielded robust methods that use weighted or probabilistic models. For example, distributional models of word meaning have been used successfully to judge paraphrase appropriateness. This has been done by representing the word meaning in context as a point in a high-dimensional semantics space (Erk and Padó, 2008; Thater et al, 2010; Erk and Padó, 2010). However, these models typically handle only individual phenomena instead of providing a meaning representation for complete sentences. It is a long-standing open question how best to integrate the weighted or probabilistic information coming from such modules with logic-based representations in a way that allows for reasoning over both. See, for example, Hobbs et al (1993).

The goal of this work is to combine logic-based meaning representations with probabilities in a single unified framework. This will allow us to obtain the best of both situations: we will have the full expressivity of first-order logic and be able to reason with probabilities. We believe that this will allow for a more complete and robust approach to natural language understanding. In order to perform logical inference with probabilities, we draw from the large and active body of work related to Statistical Relational AI (Getoor and Taskar, 2007). Specifically, we make use of Markov Logic Networks (MLNs) (Richardson and Domingos, 2006) which employ weighted graphical models to represent first-order logical formulas. MLNs are appropriate for our approach because they provide an elegant method of assigning weights to first-order logical rules, combining a diverse set of inference rules, and performing inference in a probabilistic way.

While this is a large and complex task, this paper proposes a series of first steps toward our goal. In this paper, we focus on three natural language phenomena and their interaction: implicativity and factivity, word meaning, and coreference. Our framework parses natural language into a logical form, adds rule weights computed by external NLP modules, and performs inferences using an MLN. Our end-to-end approach integrates multiple existing tools. We use Boxer (Bos et al, 2004) to parse natural language into a logical form. We use Alchemy (Kok et al, 2005) for MLN inference. Finally, we use the exemplar-based distributional model of Erk and Padó (2010) to produce rule weights.

2 Background

Logic-based semantics. Boxer (Bos et al, 2004) is a software package for wide-coverage semantic analysis that provides semantic representations in the form of Discourse Representation Structures (Kamp and Reyle, 1993). It builds on the C&C CCG parser (Clark and Curran, 2004). Bos and Markert (2005) describe a system for Recognizing Textual Entailment (RTE) that uses Boxer to convert both the premise and hypothesis of an RTE pair into first-order logical semantic representations and then uses a theorem prover to check for logical entailment.

Distributional models for lexical meaning. Distributional models describe the meaning of a word through the context in which it appears (Landauer and Dumais, 1997; Lund and Burgess, 1996), where contexts can be documents, other words, or snippets of syntactic structure. Distributional models are able to predict semantic similarity between words based on distributional similarity and they can be learned in an unsupervised fashion. Recently distributional models have been used to predict the applicability of paraphrases in context (Mitchell and Lapata, 2008; Erk and Padó, 2008; Thater et al, 2010; Erk and Padó, 2010). For example, in "The wine left a stain", "result in" is a better paraphrase for "leave" than is "entrust", while the opposite is true in "He left the children with the nurse". Usually, the distributional representation for a word mixes all its usages (senses). For the paraphrase appropriateness task, these representations are then reweighted, extended, or filtered to focus on contextually appropriate usages.

Markov Logic. An MLN consists of a set of weighted first-order clauses. It provides a way of softening first-order logic by making situations in which not all clauses are satisfied less likely but not impossible (Richardson and Domingos, 2006). More formally, let X be the set of all propositions describing a world (i.e. the set of all ground atoms), $\mathcal{F}$ be the set of all clauses in the MLN, w_i be the weight associated with clause $f_i \in \mathcal{F}$, $\mathcal{G}_{f_i}$ be the set of all possible groundings of clause f_i, and $\mathcal{Z}$ be the normalization constant. Then the probability of a particular truth assignment $\mathbf{x}$ to the variables in X is defined as:

$$P(X = \mathbf{x}) = \frac{1}{\mathcal{Z}} \exp\left(\sum_{f_i \in \mathcal{F}} w_i \sum_{g \in \mathcal{G}_{f_i}} g(\mathbf{x}) \right) = \frac{1}{\mathcal{Z}} \exp\left(\sum_{f_i \in \mathcal{F}} w_i n_i(\mathbf{x}) \right) \tag{1}$$

where $g(\mathbf{x})$ is 1 if g is satisfied and 0 otherwise, and $n_i(\mathbf{x}) = \sum_{g \in \mathcal{G}_{f_i}} g(\mathbf{x})$ is the number of groundings of f_i that are satisfied given the current truth assignment to the variables in X. This means that the probability of a truth assignment rises exponentially with the number of groundings that are satisfied.

Markov Logic has been used previously in other NLP application (e.g. Poon and Domingos (2009)). However, this paper marks the first attempt at representing deep logical semantics in an MLN.

While it is possible learn rule weights in an MLN directly from training data, our approach at this time focuses on incorporating weights computed by external knowledge sources. Weights for word meaning rules are computed from the distributional model of lexical meaning and then injected into the MLN. Rules governing implicativity and coreference are given infinite weight (hard constraints).

3 Linking logical form and vector spaces

In this section we define a link between logical form and vector space representations through a mapping function that connects predicates in logical form to points in vector space. Gärdenfors (2004) uses the interpretation function for this purpose, such that logical formulas are interpreted over vector space representations. However, he uses spaces whose dimensions are qualities, like the hue and saturation of a color or the taste of a fruit. Points in his conceptual spaces are, therefore, potential entities. In contrast, the vector spaces that we use are distributional in nature. A point in such a space is a potential word, defined through its observed contexts. For this reason, we define the link between logical form and vector space through a second mapping function independent of the interpretation function, which we call the *lexical mapping* function.

Vector space representations

[TODO:
vec is making vectors bold instead of using an overarrow]

Let V be a vector space whose dimensions stand for elements of textual context. We also write V for the set of points in the space. We assume that each word is represented as a point in vector space. [1] The central relation in vector spaces is semantic similarity. We represent this through a *similarity function*

$$S : V \times V \to [0,1]$$

that maps each pair of points in vector space to their degree of similarity. While most similarity functions in the literature are symmetric, such that $S(\mathbf{v},\mathbf{w}) = S(\mathbf{w},\mathbf{v})$, our definition also accommodates asymmmetric similarity measures like Kotlerman et al (2010). Proximity in a vector space that is distributional in nature represents substitutability in context. For that reason, the similarity function can be understood as a substitutability function. If words v and w are represented by $\mathbf{v}$ and $\mathbf{w} \in V$ and $S(\mathbf{v},\mathbf{w}) = \eta$, then w can be substituted for v to the degree η.

We follow the literature on vector space approaches to modeling word meaning in context (Erk and Padó, 2008; Thater et al, 2010; Reisinger and Mooney, 2010; Dinu and Lapata, 2010; Van de Cruys et al, 2011) in assuming that a word's context-specific meaning is a function of its out-of-context representation and the context.

[1] The assumption of a single vector per word is made for the sake of simplicity. If we want to cover models in which each word is represented through multiple vectors (Reisinger and Mooney, 2010; Dinu and Lapata, 2010), this can be done through straightforward extensions of the definitions given here.

The context may consist of a single neighboring word or multiple neighbors. The syntactic relation between the neighbors and the target word also often plays a role (Erk and Padó, 2008; Thater et al, 2010; Van de Cruys et al, 2011) .

We want to represent word meaning in a given sentence context. In previous work, Mitchell and Lapata (2008) define the meaning $\mathbf{p}$ of a two-word phrase vw as a function of the vectors for v, w, and their syntactic relation:

$$\mathbf{p} = f(\mathbf{v}, \mathbf{w}, r, K)$$

where f is some function, r is the relation between v and w in the text, and K is background knowledge. This same schema has also applied to representing the meaning of v in the presence of w (Erk and Padó, 2008). We extend this schema canonically to the case of a word v in the presence of multiple context words, also dropping the background knowledge K, since it is not clear what that would be and how it would be formalized.

We describe the context of a word v as a set $c = \{(r_1, \mathbf{w}_1), \ldots, (r_n, \mathbf{w}_n)\}$, where $\mathbf{w}_1, \ldots, \mathbf{w}_n$ are vectors in V that represent the words $w_1, \ldots, w_n$ that occur around v, and $r_i \in R$ is the semantic relation between v and w_i. Given a vector space V and a set R of semantic relations, the set $C(V,R)$ *of contexts over* V *and* R contains all finite sets of pairs from $R \times V$. Thus, $c \in C(V,R)$ for some V and some R.

We now define a function that maps the context-independent representation of a word v to its representation in a context c. A *contextualization function* on vector space V with relation set R has the form

$$\alpha : V \times C(V,R) \to V$$

For a word v in a context $c \in C(V,R)$, the meaning of v in the context c is $\alpha(\mathbf{v}, c)$.

We are particularly interested in substitutability for words in context: Given a word v in a context c, and a potential paraphrase w of v, the degree of context-specific substitutability of w for v, given their vector representations in V $\mathbf{w}$ and $\mathbf{v}$, is

$$S(\alpha(\mathbf{v}, c), \mathbf{w})$$

This formulation adapts v to the context c, but leaves the vector w unchanged, as most approaches in the literature do (Erk and Padó, 2008; Mitchell and Lapata, 2008; Thater et al, 2010; Van de Cruys et al, 2011). But we can just as well contextualize the paraphrase candidate too (Erk and Padó, 2010). We compute the degree of context-specific substitutability of w for v as

$$S(\alpha(\mathbf{v}, c), \alpha(\mathbf{w}, c))$$

This formulation contextualizes w in the same sentential context in which v is situated.

Linking logical form and vector space.

Let $\mathcal{L}$ be a logical language, a set of logical formulas. For each $n \geq 0$, let the set of n-ary predicate symbols of $\mathcal{L}$ be $\mathcal{P}^n_{\mathcal{L}}$, and let $\mathcal{P}_{\mathcal{L}} = \cup_{n\geq 0}\mathcal{P}^n_{\mathcal{L}}$. Let V be a vector space. Then a *lexical mapping* from $\mathcal{L}$ to V is a function $\ell : \mathcal{P}_{\mathcal{L}} \to V$ that maps each predicate symbol to a point in the vector space.

The aim of the lexical mapping is to be able to project inferences from vector space to logical form: If a lexical mapping function maps predicate P to $\mathbf{v}$ and Q to $\mathbf{w}$, and $S(\mathbf{v},\mathbf{w}) = \eta$, then we can substitute Q for P with certainty η. If P and Q are n-ary predicates, we can express this *weighted substitution rule* as the formula $\forall x_1,\ldots,x_n.[P(x_1,\ldots,x_n) \to Q(x_1,\ldots x_n)]$ with weight η.

Let $\mathcal{L}$ be a logical language with lexical mapping ℓ to a vector space V with similarity function S. Then the *substitution projection* for a predicate $P \in \mathcal{P}^n_{\mathcal{L}}$ is the set of weighted substitution rules (the set of pairs of formulas $F \in \mathcal{L}$ and weights $\eta \in [0,1]$) given by

$$\Pi_{S,\ell}(P) = \{(F,\eta) \mid \exists Q \in \mathcal{P}^n_{\mathcal{L}} \;[\; F = \forall x_1,\ldots,x_n.[P(x_1,\ldots,x_n) \to Q(x_1,\ldots,x_n)], \\ \eta = S\big(\ell(P),\ell(Q)\big) \;]\}$$

However, since we are interested in the substitutability of words *in context*, we compute context-specific lexical mappings by first computing a context from a logical form. Given a logical language $\mathcal{L}$, a vector space V, and set R of semantic relations, a *context mapping* is a function

$$\kappa : \mathcal{P}_{\mathcal{L}} \times \mathcal{L} \to C(V,R)$$

Given a predicate $P \in \mathcal{P}_{\mathcal{L}}$ and a formula $G \in \mathcal{L}$, it computes a context $c = \kappa(P,G)$.

By combining context mappings with contextualization functions, we can now describe how we extend a logical form by context-specific inferences: Let $\mathcal{L}$ be a logical language with lexical mapping ℓ to vector space V. Let S be a similarity function on V, α a contextualization function on V and R, and κ a context mapping from $\mathcal{L}$ to $C(V,R)$. Then the *contextualized substitution projection* for predicate $P \in \mathcal{P}^n_{\mathcal{L}}$ found in formula $G \in \mathcal{L}$ is

$$\Pi^G_{S,\ell}(P) = \{(F,\eta) \mid \exists Q \in \mathcal{P}^n_{\mathcal{L}} \;[\; F = \forall x_1,\ldots,x_n.[P(x_1,\ldots,x_n) \to Q(x_1,\ldots,x_n)], \\ \eta = S\big(\alpha(\ell(P),\kappa(P,G)),\ell(Q)\big) \;]\}$$

This ensures that similarity is measured between the replacement Q the *contextualized* vector representing P.

Thus, the aggregate contextualized substitution projection for an entire formula G is the union of the contextualized substitution projections for all predicates in G

$$\Pi^*_{S,\ell}(G) = \bigcup_{P \in \mathcal{P}_\mathcal{L} \text{ occurs in } G} \Pi^G_{S,\ell}(P)$$

This formalization only contextualizes P and estimates the substitutability of Q based on a context-independent vector. If we like,we can substitute a different lexical mapping that maps both P and Q to context-specific vectors: Given a context-independent lexical mapping ℓ, contextualization function α and context mapping κ, a predicate $P \in \mathcal{P}_\mathcal{L}$ and formula $G \in \mathcal{L}$, let $\gamma^{P,G,\alpha,\kappa}$ be the lexical mapping defined as

$$\gamma^{P,G,\ell,\alpha,\kappa}(Q) = \alpha\big(\ell(Q), \kappa(P,G)\big)$$

Then we can compute the aggregate contextualized substitution projection for G as

$$\Pi^*_{S,\ell}(G) = \bigcup_{P \in \mathcal{P}_\mathcal{L} \text{ occurs in } G} \Pi_{S,\gamma^{P,G,\ell,\alpha,\kappa}}(P)$$

4 Transforming natural language text to logical form

In transforming natural language text to logical form, we build on the software package Boxer (Bos et al, 2004). Boxer is an extension to the C&C parser (Clark and Curran, 2004) that transforms a parsed discourse of one or more sentences into a semantic representation. Boxer outputs the meaning of each discourse as a Discourse Representation Structure (DRS) that closely resembles the structures described by Kamp and Reyle (1993).

We chose to use Boxer for two main reasons. First, Boxer is a wide-coverage system that can deal with arbitrary text. Second, the DRSs that Boxer produces are close to the standard first-order logical forms that are required for use by the MLN software package Alchemy. Our system transforms Boxer output into a format that Alchemy can read and augments it with additional information.

5 Evaluation and phenomena

Textual entailment offers a good framework for testing whether a system performs correct analyses and thus draws the right inferences from a given text. For example, to test whether a system correctly handles implicative verbs, one can use the *premise* *p* along with the *hypothesis* *h* in (1) below. If the system analyses the two sentences correctly, it should infer that *h* holds. While the most prominent forum using textual entailment is the Recognizing Textual Entailment (RTE) challenge (Dagan et al, 2005), the RTE datasets do not test the phenomena in which we are interested. For example, in order to evaluate our system's ability to determine word meaning in context, the RTE pair would have to specifically test word sense confusion by having a word's context in the hypothesis be different from the context of the premise. However, this simply does not occur in the RTE corpora. In order to properly test our phenomena, we construct hand-tailored premises and hypotheses based on real-world texts.

In this paper, we focus on three natural language phenomena and their interaction: implicativity and factivity, word meaning, and coreference. The first phenomenon, implicativity and factivity, is concerned with analyzing the truth conditions of nested propositions. For example, in the premise of the entailment pair shown in example (1), "arrange that" falls under the scope of "forget to" and "fail" is under the scope of "arrange that". Correctly recognizing nested propositions is necessary for preventing false inferences such as the one in example (2).

Example 1. *p*: Ed did not forget to arrange that Dave fail[2]

h: Dave failed

Example 2. *p*: The mayor hoped to build a new stadium[3]

*h**:The mayor built a new stadium

For the second phenomenon, word meaning, we address paraphrasing and hypernymy. For example, in (3) "covering" is a good paraphrase for "sweeping" while "brushing" is not.

Example 3. *p*: A stadium craze is **sweeping** the country

h_1:A stadium craze is **covering** the country

h_2*:A stadium craze is **brushing** the country

The third phenomenon is coreference, as illustrated in (4). For this example, to correctly judge the hypothesis as entailed, it is necessary to recognize that "he" corefers with "Christopher" and "the new ballpark" corefers with "a replacement for Candlestick Park".

Example 4. *p*: George Christopher has been a critic of the plan to build a replacement for Candlestick Park. As a result, he won't endorse the new ballpark.

h: Christopher won't endorse a replacement for Candlestick Park.

[2] Examples (1) and (7) and Figure **??** are based on examples by MacCartney and Manning (2009)

[3] Examples (2), (3), (4), and (9) are modified versions of sentences from document wsj_0126 from the Penn Treebank

Some natural language phenomena are most naturally treated as categorial, while others are more naturally treated using weights or probabilities. In this paper, we treat implicativity and coreference as categorial phenomena, while using a probabilistic approach to word meaning.

6 Evaluation

As a preliminary evaluation of our system, we constructed a set of demonstrative examples to test our ability to handle the previously discussed phenomena and their interactions and ran each example with both a theorem prover and Alchemy. Note that when running an example in the theorem prover, weights are not possible, so any rule that would be weighted in an MLN is simply treated as a "hard clause" following Bos and Markert (2005).

Checking the logical form. We constructed a list of 72 simple examples that exhaustively cover cases of implicativity (positive, negative, null entailments in both positive and negative environments), hypernymy, quantification, and the interaction between implicativity and hypernymy. The purpose of these simple tests is to ensure that our flattened logical form and truth condition rules correctly maintain the semantics of the underlying DRSs. Examples are given in (5).

Example 5. (a)The mayor did not manage to build a stadium $\not\models$ The mayor built a stadium

(b)Fido is a dog and every dog walks $\models$ A dog walks

Examples in previous sections. Examples (1), (2), (3), (**??**), and (**??**) all come out as expected. Each of these examples demonstrates one of the phenomena in isolation. However, example (4) returns "not entailed", the incorrect answer. As discussed previously, this failure is a result of our system's inability to correctly incorporate the complex coreferring expression "a replacement for Candlestick Park". However, the system *is* able to correctly incorporate the coreference of "he" in the second sentence to "Christopher" in the first.

Implicativity and word sense. For example (6), "fail to" is a negatively entailing implicative in a positive environment. So, p correctly entails h_{good} in both the theorem prover and Alchemy. However, the theorem prover incorrectly licenses the entailment of h_{bad} while Alchemy does not. The probabilistic approach performs better in this situation because the categorial approach does not distinguish between a good paraphrase and a bad one. This example also demonstrates the advantage of using a context-sensitive distributional model to calculate the probabilities of paraphrases because "reward" is an *a priori* better paraphrase than "observe" according to WordNet since it appears in a higher ranked synset.

Example 6. p: The U.S. is watching closely as South Korea fails to honor U.S. patents[4]

h_{good} South Korea does not **observe** U.S. patents

h_{bad} South Korea does not **reward** U.S. patents

Implicativity and hypernymy. MacCartney and Manning (2009) extended the work by Nairn et al (2006) in order to correctly treat inference involving monotonicity and exclusion. Our approaches to implicatives and factivity and hyper/hyponymy combine naturally to address these issues because of the structure of our logical representations and rules. For example, no additional work is required to license the entailments in (7).

Example 7. (a) John refused to dance ⊨ John didn't tango

(b) John did not forget to tango ⊨ John danced

Example (8) demonstrates how our system combines categorial implicativity with a probabilistic approach to hypernymy. The verb "anticipate that" is positively entailing in the negative environment. The verb "moderate" can mean "chair" as in "chair a discussion" or "curb" as in "curb spending". Since "restrain" is a hypernym of "curb", it receives a weight based on the applicability of the word "curb" in the context. Similarly, "talk" receives a weight based on its hyponym "chair". Since our model predicts "curb" to be a more probable paraphrase of "moderate" in this context than "chair" (even though the priors according to WordNet are reversed), the system is able to infer h_{good} while rejecting h_{bad}.

Example 8. p: He did not anticipate that inflation would moderate this year

h_{good} Inflation **restrained** this year

h_{bad} Inflation **talked** this year

Word sense, coreference, and hypernymy. Example (9) demonstrates the interaction between paraphrase, hypernymy, and coreference incorporated into a single entailment. The relevant coreference chains are marked explicitly in the example. The correct inference relies on recognizing that "he" in the hypothesis refers to "Joe Robbie" and "it" to "coliseum", which is a hyponym of "stadium". Further, our model recognizes that "sizable" is a better paraphrase for "healthy" than "intelligent" even though WordNet has the reverse order.

Example 9. p: [Joe Robbie]$_{53}$ couldn't persuade the mayor , so [he]$_{53}$ built [[his]$_{53}$ own coliseum]$_{54}$.

[4] Example (6) is adapted from Penn Treebank document wsj_0020 while (8) is adapted from document wsj_2358

[He]$_{53}$ has used [it]$_{54}$ to turn a healthy profit.[5]

h_{good}: *Joe Robbie* used *a stadium* to turn a **sizable** profit

h_{bad1}: *Joe Robbie* used *a stadium* to turn an **intelligent** profit

h_{bad2}: *The mayor* used *a stadium* to turn a healthy profit

7 Future work

[pull out realations from the logical form to have more interesting α]

[think about vector spaces that take logical form into account similar to how pado and lapata took dependencies into account.]

[mention vibhav's work]

[mention vectors built form entire sentences. can measure similarity between phrases.]

The next step is to execute a full-scale evaluation of our approach using more varied phenomena and naturally occurring sentences. However, the memory requirements of Alchemy are a limitation that prevents us from currently executing larger and more complex examples. The problem arises because Alchemy considers every possible grounding of every atom, even when a more focused subset of atoms and inference rules would suffice. There is on-going work to modify Alchemy so that only the required groundings are incorporated into the network, reducing the size of the model and thus making it possible to handle more complex inferences. We will be able to begin using this new version of Alchemy very soon and our task will provide an excellent test case for the modification.

Since Alchemy outputs a probability of entailment, it is necessary to fix a threshold that separates entailment from nonentailment. We plan to use machine learning techniques to compute an appropriate threshold automatically from a calibration dataset such as a corpus of valid and invalid paraphrases.

8 Conclusion

In this paper, we have introduced a system that implements a first step towards integrating logical semantic representations with probabilistic weights using methods from Statistical Relational AI, particularly Markov Logic. We have focused on three

[5] Only relevent coreferences have been marked

phenomena and their interaction: implicatives, coreference, and word meaning. Taking implicatives and coreference as categorial and word meaning as probabilistic, we have used a distributional model to generate paraphrase appropriateness ratings, which we then transformed into weights on first order formulas. The resulting MLN approach is able to correctly solve a number of difficult textual entailment problems that require handling complex combinations of these important semantic phenomena.

Acknowledgements

This work was supported by the Department of Defense (DoD) through a National Defense Science and Engineering Graduate Fellowship (NDSEG) Fellowship for the first author, National Science Foundation grant IIS-0845925 for the second author, and a grant from the Longhorn Innovation Fund for Technology. [TODO: Is all of this right?]

References

Bos J, Markert K (2005) Recognising textual entailment with logical inference. In: Proceedings of EMNLP 2005, pp 628–635

Bos J, Clark S, Steedman M, Curran JR, Hockenmaier J (2004) Wide-coverage semantic representations from a CCG parser. In: Proceedings of COLING 2004, Geneva, Switzerland, pp 1240–1246

Clark S, Curran JR (2004) Parsing the WSJ using CCG and log-linear models. In: Proceedings of ACL 2004, Barcelona, Spain, pp 104–111

Dagan I, Glickman O, Magnini B (2005) The pascal recognising textual entailment challenge. In: In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment

Dinu G, Lapata M (2010) Measuring distributional similarity in context. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Cambridge, MA, pp 1162–1172, URL `http://www.aclweb.org/anthology/D10-1113`

Erk K, Padó S (2008) A structured vector space model for word meaning in context. In: Proceedings of EMNLP 2008, Honolulu, HI, pp 897–906

Erk K, Padó S (2010) Exemplar-based models for word meaning in context. In: Proceedings of ACL 2010, Uppsala, Sweden, pp 92–97

Gärdenfors P (2004) Conceptual spaces. MIT press, Cambridge, MA

Getoor L, Taskar B (eds) (2007) Introduction to Statistical Relational Learning. MIT Press, Cambridge, MA

Hobbs JR, Stickel M, Appelt D, Martin P (1993) Interpretation as abduction. Artificial Intelligence 63(1–2):69–142

Kamp H, Reyle U (1993) From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT. Kluwer, Dordrecht

Kok S, Singla P, Richardson M, Domingos P (2005) The Alchemy system for statistical relational AI. Tech. rep., Department of Computer Science and Engineering, University of Washington, `http://www.cs.washington.edu/ai/alchemy`

Kotlerman L, Dagan I, Szpektor I, Zhitomirsky-Geffet M (2010) Directional distributional similarity for lexical inference. Natural Language Engineering 16(04):359–389

Landauer T, Dumais S (1997) A solution to Platos problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review 104(2):211–240

Lund K, Burgess C (1996) Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior Research Methods, Instruments, and Computers 28:203—208

MacCartney B, Manning CD (2009) An extended model of natural logic. In: Proceedings of the Eighth International Conference on Computational Semantics (IWCS-8), pp 140–156

Mitchell J, Lapata M (2008) Vector-based models of semantic composition. In: Proceedings of ACL, pp 236–244

Nairn R, Condoravdi C, Karttunen L (2006) Computing relative polarity for textual inference. In: Proceedings of Inference in Computational Semantics (ICoS-5), Buxton, UK

Poon H, Domingos P (2009) Unsupervised semantic parsing. In: Proceedings of EMNLP 2009, pp 1–10

Reisinger J, Mooney RJ (2010) Multi-prototype vector-space models of word meaning. In: Proceeding of NAACL

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

Thater S, Fürstenau H, Pinkal M (2010) Contextualizing semantic representations using syntactically enriched vector models. In: Proceedings of ACL 2010, Uppsala, Sweden, pp 948–957

Van de Cruys T, Poibeau T, Korhonen A (2011) Latent vector weighting for word meaning in context. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK., pp 1012–1022