On the Proper Treatment of Quantifiers in Probabilistic Logic Semantics

Islam Beltagy
The University of Texas at Austin
Computer Science Department
beltagy@cs.utexas.edu

Katrin Erk
The University of Texas at Austin
Linguistics Department
katrin.erk@mail.utexas.edu

Abstract

As a format for describing the meaning of natural language sentences, probabilistic logic combines the expressivity of first-order logic with the ability to handle graded information in a principled fashion. But practical probabilistic logic frameworks usually assume a finite domain in which each entity corresponds to a constant in the logic (domain closure assumption). They also assume a closed world where everything has a very low prior probability. These assumptions lead to some problems in the inferences that these systems make. In this paper, we show how to formulate Textual Entailment (RTE) inference problems in probabilistic logic in a way that takes the domain closure and closed-world assumptions into account. We evaluate our proposed technique on three RTE datasets, on a synthetic dataset with a focus on complex forms of quantification, on FraCas and on one more natural dataset. We show that our technique leads to improvements on the more natural dataset, and achieves 100% accuracy on the synthetic dataset and on the relevant part of FraCas.

1 Introduction

Tasks in natural language semantics are becoming more fine-grained, like Textual Entailment (Dagan et al., 2013), Semantic Parsing (Kwiatkowski et al., 2013; Berant et al., 2013), or fine-grained opinion analysis. With the more complex tasks, there has been a renewed interest in phenomena like negation (Choi and Cardie, 2008) or the factivity of embedded clauses (MacCartney and Manning, 2009; Lotan et al., 2013) – phenomena that used to be standardly handled by logic-based semantics. Bos (2013) identifies the use of broad-coverage lexical resources as one aspect that is crucial to the success of logic-based approaches. Another crucial aspect is the ability to reason with uncertain, probabilistic information (Garrette et al., 2011; Beltagy et al., 2013). Lexical information typically comes with weights, be it weights of paraphrase rules (Lin and Pantel, 2001; Ganitkevitch et al., 2013), confidence ratings of word sense disambiguation systems, or distributional similarity values, and reasoning with such information and finding the overall best interpretation requires the ability to handle weights. This is possible in the framework of probabilistic logic (Nilsson, 1986).

In this paper we do not talk about *why* one should use probabilistic logic for natural language semantics (we argue for the need for probabilistic logic in previous work (Beltagy et al., 2013) which is summarized in section 2.4), we focus on the *how*, as it turns out that some practical design properties of probabilistic reasoning systems make it necessary to make changes to the meaning representations. One characteristic of practical probabilistic logic frameworks such as Markov Logic (Richardson and Domingos, 2006) is that they assume a finite domain, in particular they assume that the entities in the domain correspond to the constants mentioned in the set of formulas at hand. This is the *domain closure assumption (DCA)*, a strong assumption that reduces any inference problem to the propositional case. It also has far-reaching consequences on the behavior of a system. For example, suppose we know that *Tweety is a bird that flies*: $bird(T) \wedge fly(E) \wedge agent(T, E)$ (There is a flying event of which Tweety is the agent. We use this Neo-Davidsonian representation throughout the paper, as it is also produced by the wide-coverage semantic analysis system we use). Then we can conclude that *every bird flies*, because

Proceedings of the 11th International Conference on Computational Semantics, pages 140–150,
London, UK, April 15-17 2015.

by the DCA we are only considering models with a domain of size one. Of that single entity, we know both that it is a bird and that it flies. In a natural language inference setting, such as Textual Entailment, this is not the conclusion we would like to draw. So we need to use a nonstandard encoding to ensure that existential and universal quantifiers behave in the way they should.

Another issue that we face is that practical probabilistic logic frameworks usually have to construct all groundings of the given formulas before doing inference. The *closed-world assumption (CWA)* – the assumption that nothing is the case unless stated otherwise – helps to keep memory use for this grounding step in check (Beltagy and Mooney, 2014). However, the CWA comes with inference problems of its own, for example it would let us infer from *The sky is blue* that *No girl is dancing* because by the CWA we assume that no entity is dancing unless we were told otherwise.

In this paper, we concentrate on entailment, one of the fundamental criteria of language understanding. We show how to formulate probabilistic logic inference problems for the task of Recognizing Textual Entailment (RTE, Dagan et al. (2013)) in a way that takes the domain closure assumption as well as the closed-world assumption into account. We evaluate our approach on three RTE datasets. The first is a synthetic dataset that exhaustively tests inference performance on sentences with two quantifiers. We get 100% accuracy on this dataset. We also evaluate on the first section of the FraCas dataset (Cooper et al., 1996), a collection of Textual Entailment problems tailored focusing on particular semantic phenomena. We restrict our analysis to sentences with determiners that our current system handles (excluding "few", "most", "many" and "at least"), and we get 100% accuracy on them. Also, we evaluate on the RTE part of the SICK dataset (Marelli et al., 2014) and show that our approach leads to improvements.

2 Background and Related Work

2.1 Recognizing Textual Entailment

Recognizing Textual Entailment (RTE) (Dagan et al., 2013) is the task of determining whether one natural language text, the *Text* T, *Entails*, *Contradicts*, or is *Neutral* with respect to another, the *Hypothesis* H. Here are examples from the SICK dataset (Marelli et al., 2014):

- Entailment

T: A man and a woman are walking together through the woods.
H: A man and a woman are walking through a wooded area.

- Contradiction

T: A man is jumping into an empty pool
H: A man is jumping into a full pool

- Neutral

T: A young girl is dancing
H: A young girl is standing on one leg

2.2 Statistical relational learning

Statistical Relational Learning (SRL) techniques (Getoor and Taskar, 2007) combine logical and statistical knowledge in one uniform framework and provide a mechanism for coherent probabilistic inference. They typically employ weighted formulas in first-order logic to compactly encode complex probabilistic graphical models. Weighted rules allow situations in which not all clauses are satisfied. These frameworks typically operate on the set of all groundings of a given set of formulas. A probabilistic logic program defines a probability distribution over the possible values of the ground atoms where they are treated as random variables. In addition to a set of rules R, a probabilistic logic program takes an evidence set E asserting some truth values about some of the random variables. Then, given a query formula Q, probabilistic logic inference calculates the probability $P(Q|R, E)$ which is the answer to the query.

2.3 Markov Logic Networks

Markov Logic Networks (MLN) (Richardson and Domingos, 2006) are one of the statistical relational learning frameworks. MLNs work as templates to build graphical models that define probability distributions over worlds (equivalently, truth assignments to ground atoms), where a world's probability increases exponentially with the total weight of the ground clauses that it satisfies. Probability of a given world x is denoted by:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right) \quad (1)$$

where Z is the partition function, i ranges over all formulas F_i is the MLN, w_i is the weight of F_i and $n_i(x)$ is the number of true groundings of F_i in the world x. The marginal inference of MLNs calculates the probability $P(Q|E, R)$, where Q is a query, E is the evidence set, and R is the set of weighted formulas.

Alchemy (Kok et al., 2005) is the most widely used MLN implementation. It is a software package that contains implementations of a variety of MLN inference and learning algorithms. However, developing a scalable, general-purpose, accurate inference method for complex MLNs is an open problem.

2.4 Markov Logic Networks for Natural Language Semantics

Logic-based natural language semantics follows the framework of Montague Grammar as laid out in Montague's "Proper Treatment of Quantification in Ordinary English" (Montague, 1973). Recent wide-coverage tools that use logic-based sentence representations include Copestake and Flickinger (2000), Bos (2008), and Lewis and Steedman (2013).

In this work, we build on the approach of Beltagy et al. (2013) and Beltagy and Mooney (2014). Beltagy et al. (2013) use Bos' system Boxer to map English sentences to logical form, then use Markov Logic Networks to reason over those meaning representations, evaluating on two tasks, RTE and sentence similarity (semantic textual similarity, STS (Agirre et al., 2012)). Weighted clauses are used to encode distributional similarity of words or phrases as a set KB of weighted inference rules. An RTE problem of whether Text T entails Hypothesis H given the set of inference rules KB is transformed into the question of the probability of the hypothesis given the text and the rules, $P(H|T, KB)$. The Alchemy tool (Kok et al., 2005) is used to estimate this probability. Then a classifier is trained to convert the probabilities into judgments of Entailment, Contradiction, or Neutral. However, because of lack of adaptation to domain closure and other problems, the approach of Beltagy et al. (2013) is not able to make use of universal quantifiers in the Hypothesis H. In this paper, we address this problem.

Beltagy and Mooney (2014) propose an inference algorithm that can solve the inference problem $P(H|T, KB)$ efficiently for complex queries. They enforce a closed-world assumption that significantly reduces the problem size. However, this closed-world assumption makes negated hypotheses H come true regardless of T. In this paper, we also address this issue.

3 Approach

A significant difference between standard logic and probabilistic logic comes from the fact that practical probabilistic logic frameworks typically make the Domain Closure Assumption (DCA, Genesereth and Nilsson (1987); Richardson and Domingos (2006)): The only models considered for a set F of formulas are those for which the following three conditions hold. (a) Different constants refer to different objects in the domain, (b) the only objects in the domain are those that can be represented using the constant and function symbols in F, and (c) for each function f appearing in F, the value of f applied to every possible tuple of arguments is known, and is a constant appearing in F. Together, these three conditions entail that *there is a one-to-one relation between objects in the domain and the named constants of* F.

For working with Markov Logic Networks in practice, this means that constants need to be explicitly introduced in the probabilistic logic program. Constants are used to ground the predicates, and build the

MLN's graphical model. Different sets of constants result into different graphical models. If no constants are explicitly introduced, the graphical model is empty (no random variables). A more serious problem is that the DCA affects the behavior of universal quantifiers, as illustrated with the Tweety example in the introduction. We address this problem by adding more constants to the system.

While the DCA is not commonly made outside of probabilistic logic systems, the Closed World Assumption (CWA) is widely used in inference systems. However, in the context of Textual Entailment, it, too, leads to problems, as it assumes that all negative information is true a priori – but in RTE we do not want to assume anything to be true unless stated in the Text, neither positive nor negative information.

This section discusses the changes that we make to the representations of natural language sentence meaning in order to deal with the DCA and CWA. We focus on the Textual Entailment task as these changes will be different for the evidence given (the Text T) and the query (the Hypothesis H).

3.1 Skolemization

Skolemization (Skolem, 1920) transforms a formula $\forall x_1 \ldots x_n \exists y.F$ to $\forall x_1 \ldots x_n.F^*$, where F^* is formed from F by replacing all free occurrences of y by a term $f(x_1, \ldots, x_n)$ for a new function symbol f. If $n = 0$, f is called a *Skolem constant*, otherwise a *Skolem function*. Although Skolemization is a widely used technique when using standard first-order logic, it is typically not used in the context of probabilistic logic because typical probabilistic logic applications do not require existential quantifiers. In addition, the standard way of dealing with an existential quantifier in probabilistic logic is by replacing it with a disjunction over all constants in the domain (Richardson and Domingos, 2006). In our case, Skolemization plays a role for existential quantifiers in the text T. Take for example *T: A man is driving a car*, which in logical form is

$$T : \exists x, y, z.\ man(x) \wedge agent(y, x) \wedge drive(y) \wedge patient(y, z) \wedge car(z)$$

Because of the DCA, we need to explicitly introduce constants into the domain. Skolemization solves this problem. Non-embedded existentially quantified variables like in the example above are replaced with Skolem constants

$$T : man(M) \wedge agent(D, M) \wedge drive(D) \wedge patient(D, C) \wedge car(C)$$

where M, D, C are constants introduced into the domain. Standard Skolemization would replace an embedded existentially quantified variable y with a Skolem function depending on all universally quantified variables x under which y is embedded. For example, here is the logical form of *T: All birds fly*.

$$T : \forall x.\ bird(x) \Rightarrow \exists y.\ agent(y, x) \wedge fly(y)$$

It is Skolemized as follows:

$$T : \forall x.\ bird(x) \Rightarrow agent(f(x), x) \wedge fly(f(x))$$

By condition (c) of the DCA, if we used a Skolem function, it would need to map its argument to a constant. In particular, it would need to map each argument to a new constant to state that for every known bird (i.e., for any constant that is a bird) there is a separate flying event. To achieve this, we introduce a new predicate $skolem_f$ that we use instead of the Skolem function f, and for every constant that is a bird, we add an extra constant that is a flying event. The example above then becomes

$$T : \forall x.\ bird(x) \Rightarrow \forall y.\ skolem_f(x, y) \Rightarrow agent(y, x) \wedge fly(y)$$

Assume that we have evidence of a single bird B_1. Then we introduce a new constant C_1 and an atom $skolem_f(B_1, C_1)$ to simulate that the Skolem function f maps the constant B_1 to the constant C_1.

3.2 Existence

Suppose we have a sentence *T: All birds with wings fly*. Its representation will yield an empty graphical model because there are no constants in the system. However, pragmatically this sentence presupposes that there are, in fact, birds with wings (Strawson, 1950; Geurts, 2007). By default, probabilistic logic and standard first-order logic do not capture this existential presupposition. We add it here to avoid the problem of empty graphical models. In a simplification of the account of Geurts (2007), we assume that the domain of almost all quantifiers is presupposed to be nonempty.

"Existence" deals with universal quantifiers in the text T. Each universal quantifier $all(restrictor, body)$ has a body and restrictor. From the parse tree of a sentence, bodies and restrictors of each quantifier can be identified. We add an existence rule for the entities in the restrictor of each universal quantifier. For example, *T: All birds with wings fly*,

$$T : \forall x, y.\, bird(x) \wedge with(x, y) \wedge wing(y) \Rightarrow \exists z.\, agent(z, x) \wedge fly(z)$$

is changed to *T: All birds with wings fly, and there is a bird with wings*.

$$T : (\, \forall x, y.\, bird(x) \wedge with(x, y) \wedge wing(y) \Rightarrow \exists z.\, agent(z, x) \wedge fly(z)\,) \\ \wedge (\, \exists u, v.\, bird(u) \wedge with(u, v) \wedge wing(v)\,)$$

Then we leave it to the Skolemization to generate constants and evidence representing the *bird with wings*, $bird(B) \wedge with(B, W) \wedge wing(W)$.

Here is another example, for a universal quantifier that comes from a negated existential, *T: No bird flies*, which in logic is:

$$T : \neg\exists x, y.\, bird(x) \wedge agent(y, x) \wedge fly(y)$$

or equivalently (by rewriting it as restrictor and body)

$$T : \forall x, y.\, bird(x) \Rightarrow \neg(fly(y) \wedge agent(y, x))$$

The Existence assumption is applied to the restrictor *bird*, so it modifies this sentence to *T: No bird flies, and there is a bird*.

$$T : (\, \neg\exists x, y.\, bird(x) \wedge agent(y, x) \wedge fly(y)\,) \wedge (\, \exists v.\, bird(v)\,)$$

Again, Skolemization generates the constants and evidence $bird(B)$.

One special case that we need to take into consideration is sentences like *T: There are no birds*, which in logic is

$$T : \neg\exists x.\, bird(x)$$

Although $bird(x)$ has a universally quantified variable, we do not generate an existence rule for it. In this case the nonemptiness of the domain is not assumed because the sentence explicitly negates it.

3.3 Universal quantifiers in the hypothesis

Under the DCA, it is possible to conclude from *T: Tweety is a bird that flies* that *H: all birds fly*, as discussed above, because H is true for all constants *in the domain*. While we used Skolemization and Existence to handle issues in the representation of T, this problem affects universally quantified variables in H. Similar to what we do for universal quantifiers in T, we introduce new constants to handle universal quantifiers in H, but for a different rationale.

Consider T_1*: There is a black bird*, T_2*: All birds are black*, and *H: All birds are black.* These sentences are represented as

$$T_1 : \exists x.\, bird(x) \wedge black(x)$$
$$Skolemized\, T_1 : bird(B) \wedge black(B)$$

$$T_2 : \forall x.\, bird(x) \Rightarrow black(x)$$
$$Skolemized\, T_2 : \forall x.\, bird(x) \Rightarrow black(x)$$

$$H : \forall x.\, bird(x) \Rightarrow black(x)$$

We want H to be judged true only if there is evidence that all birds will be black, no matter how many birds there are in the domain, as is the case in T_2 but not T_1. So we introduce a new constant D and assert $bird(D)$ to test if it follows that $black(D)$. The new evidence $bird(D)$ prevents the hypothesis from being judged true given T_1. Given T_2, the new bird D will be inferred to be black, in which case we take the hypothesis to be true.[1]

As with Existence, the same special case need to be taken into consideration. For sentences like *H: There are no birds*, which in logic is

$$H : \neg\exists x.\, bird(x)$$

we do not generate any hard evidence for $bird(x)$.

3.4 Negative hypotheses and the closed-world assumption

As discussed above, practical probabilistic logic systems typically operate on ground formulas, and the grounding step can require significant amounts of memory. Making the closed-world assumption (CWA) mitigates this effect (Beltagy and Mooney, 2014). In a probabilistic logic system, the CWA takes the form of a statement that everything has a very low prior probability. The problem here is that a negated hypothesis H could come true just because of the CWA, not because the negation is explicitly stated in T. Here is an example that demonstrates the problem, *H: There is no young girl dancing*:

$$H : \forall x, y.\, young(x) \wedge girl(x) \Rightarrow \neg(agent(y, x) \wedge dance(y))$$

As in section 3.3, we generate from H evidence of a young girl $young(G) \wedge girl(G)$ because it is in the restrictor of the universal quantifier (stating that an additional *young girl* exists in the world in general, not in this particular situation). For H to be true, inference needs to find out that G is not dancing. We need H to be true only if T is explicitly negating the existence of a young girl that dances, but because of the CWA, H could become true even if T is uninformative.

To make sure that H becomes true only if it is entailed by T, we construct a new rule R which, together with the evidence generated from H, states the opposite of the negated parts of H. R is formed as a conjunction of all the predicates that were not used to generate evidence before, and are *negated* in H. This rule R gets a positive weight indicating that its ground atoms have high prior probability. This way, the prior probability of H is low, and H cannot become true unless T explicitly negates R. Here is a Neutral RTE example adapted from the SICK dataset, *T: A young girl is standing on one leg*, and *H: There is no young girl dancing*. Their representations are

T: $\exists x, y, z.\, young(x) \wedge girl(x) \wedge agent(y, x) \wedge stand(y) \wedge on(y, z) \wedge one(z) \wedge leg(z)$
H: $\forall x, y.\, young(x) \wedge girl(x) \Rightarrow \neg(agent(y, x) \wedge dance(y))$
E: $young(G) \wedge girl(G)$
R: $agent(D, G) \wedge dance(D)|w = 1.5$

[1]Note that this strategy can fail, for example given *T: There is a black bird, and if there are exactly two birds, then all birds are black.* If we were to handle quantifiers like *exactly two*, which we are not doing yet, we would mistakenly conclude that all birds are black. However, we expect that sentences like this will be extremely rare in naturally occurring text.

E is the evidence generated for the restrictor of the universal quantifier in H, and R is the weighted rule for the remaining negated predicates. The relation between T and H is Neutral, as T does not entail H. This means, we want $P(H|T, E) = 0$, but because of the CWA, $P(H|T, E) \approx 1$. Adding R solves this problem and $P(H|T, E, R) \approx 0$ because H is not explicitly entailed by T.

In case H contains existentially quantified variables that occur in negated predicates, they need to be universally quantified in R for H to be false by default. For example, ***H: There is some bird that is not black***:

$$H : \exists x.\, bird(x) \wedge \neg black(x)$$
$$R : \forall x.\, black(x) | w = 1.5$$

Without R, the prior probability of H is $P(H) \approx 1$ because by default any bird is not black. However, with R, $P(H|R) \approx 0$. If one variable is universally quantified and the other is existentially quantified, we need to do something more complex. Here is an example, ***H: The young girl is not dancing***:

$$H : \exists x.\, young(x) \wedge girl(x) \wedge \neg(\, \exists y.\, agent(y, x) \wedge dance(y)\,)$$
$$R : \forall v.\, agent(D, v) \wedge dance(D) | w = 1.5$$

If H is a negated formula that is entailed by T, then T (which has infinite weight) will contradict R, allowing H to be true. Any weighted inference rules in KB will need weights high enough to overcome R. So the weight of R is taken into account when computing inference rule weights.

4 Evaluation

We evaluate on three RTE datasets. The first is a synthetic dataset that exhaustively tests inference performance on sentences with two quantifiers. The second is the RTE part of the SICK dataset (Marelli et al., 2014). The third is FraCas (Cooper et al., 1996).

4.1 Synthetic dataset

We automatically generate an RTE dataset that exhaustively test inferences on sentences with two quantifiers. Each RTE pair (T, H) is generated following this format:

$$T : Q_{t1}(L_{t1}, Q_{t2}(L_{t2}, R_{t2}))$$
$$H : Q_{h1}(L_{h1}, Q_{h2}(L_{h2}, R_{h2}))$$

where

- $Q_x \in$ {some, all, no, not all}
- $L_{t1}, L_{h1} \in$ {man, hungry man} and $L_{t1} \neq L_{h1}$
- $L_{t2}, L_{h2} \in$ {food, delicious food} and $L_{t2} \neq L_{h2}$
- $R_{t2}, R_{h2} =$ eat

Informally, the dataset has all possible combinations of sentences with two quantifiers. Also it has all possible combinations of monotonicity directions – upward and downward – between L_{t1} and L_{h1} and between L_{t2} and L_{h2}. The dataset size is 1,024 RTE pairs. Here is an example of a generated RTE pair:

T: No man eats all food

H: Some hungry men eat not all delicious food

The dataset is automatically annotated for entailment decisions by normalizing the logical forms of the sentences and then using standard monotonicy rules on the bodies and restrictors of the quantifiers. 72 pairs out of the 1,024 are entailing pairs, and the rest are non-entailing.

Our system computes $P(H|T)$. The resulting probability between 0 and 1 needs to be mapped to an Entail/Non-entail decision. In this dataset, and because we do not have weighted inference rules, all output probabilities close to 1 denote Entail and probabilities close to 0 denote Non-entail.

	Synthetic dataset			SICK dataset	FraCas dataset	
System	Accuracy	FP	FN	Accuracy	Gold parses	System parses
Baseline(most common class)	92.96%	0	72	56.36%	47.82%	47.82%
Skolem	50.78%	472	32	68.10%	50.00%	43.48%
Skolem + Existence	57.03%	440	0	68.10%	43.48%	36.96%
Skolem + ($\forall$ in H)	82.42%	140	40	68.14%	63.04%	50.00%
Skolem + ($\forall$ in H) + CWA	96.09%	0	40	76.48%	100.0%	84.78%
Full system	**100%**	0	0	**76.52%**	**100.0%**	84.78%

Table 1: Results of all datasets on different configurations of the system. The most common class baseline is Non-entail for the synthetic dataset, Neutral for SICK and Entail for FraCas. False positives (FP) and False negatives (FN) statistic are reported only for the synthetic dataset because it is a binary classification task. FP/FN results are counts out of 1,024.

Results The leftmost part of Table 1 summarizes the results on the synthetic dataset in terms of accuracy. The baseline always judges non-entailment. Ablation tests are as follows. *Skolem* is a system that applies Skolemization to existentially quantified variables in the Text T (Sec. 3.1) but none of the other adaptations of Section 3. *Existence* is a system that makes the existence assumption for universal quantifiers in T (Sec. 3.2). *($\forall$ in H)* is constant introduction for universal quantifiers in the Hypothesis (Sec. 3.3). Finally, *CWA* is a system that handles negation in the Hypothesis H in a way that takes the closed-world assumption into account (Sec. 3.4). The results in Table 1 show the importance of each part of the proposed system. Skolemization and the Existence assumption eliminate some false negatives from missing constants. All false positives are eliminated when constants are introduced for universal quantifiers in the Hypothesis ($\forall$ in H) and when the effects of the closed-world assumption are counteracted (*CWA*). The full system achieves 100% accuracy, showing that our formulation is perfectly handling the DCA and CWA on these complex quantified sentences.

Note that a pure logic system such as Boxer (Bos, 2008) can also achieve 100% on this dataset. But again, the aim of this paper is not to argue that probabilistic logic is preferable to standard first order logic. This can only be done by showing that weighted, uncertain knowledge leads to better inferences. Rather, this paper constitutes a necessary prerequisite, in that it is necessary first to determine how to achieve the correct (hard) inferences with probabilistic logic before we measure the effect of weighting.

4.2 The SICK dataset

Sentences Involving Compositional Knowledge (SICK, Marelli et al. (2014)) is an RTE dataset collected for the SemEval 2014 competition. It consists of 5,000 T/H pairs for training and 5,000 for testing. Pairs are annotated for both RTE and STS (sentence similarity). For the purpose of this paper, we only use the RTE annotation.

Pairs in SICK (as well as FraCas in the next section) are classified into three classes, Entailment, Contradiction, and Neutral. This means that computing the probability $P(H|T)$ alone is not enough for this threeway classification. We additionally compute $P(\neg T|H)$. Entailing pairs have $P(H|T) \approx 1$, $P(\neg T|H) \approx 0$, Contradicting pairs have $P(H|T) \approx 0$, $P(\neg T|H) \approx 1$, and Neutral pairs have $P(H|T) \approx P(\neg T|H)$. We use these two conditional probabilities as input features to an SVM classifier trained on the training set to mapped them to an Entail/Neutral/Contradict decision.

Results The middle panel of Table 1 reports results on the SICK dataset, again in terms of accuracy. Almost all sentences in the SICK dataset are simple existentially quantified sentences except for a few sentences with an outer negation. Accordingly, the system with Skolemization basically achieves the same accuracy as when Existence and ($\forall$ in H) are added. Handling negation in H effectively improves the accuracy of our system by reducing the number of false positives resulting from the CWA.

4.3 The FraCas dataset

FraCas (Cooper et al., 1996)[2] is a dataset of hand-built entailments pairs. The dataset consists of 9 sections, each of which is testing a different set of phenomena. For this paper, we use sentences from the first section, which tests quantification and monotonicity. However, we exclude pairs containing the determiners "few", "most", "many" and "at least" because our system does not currently have a representation for them. We evaluate on 46 pairs out of 74.[3][4] Because of that, we cannot compare with previous systems that evaluate on the whole section (MacCartney and Manning, 2008; Lewis and Steedman, 2013).

To map sentences to logical form, we use Boxer as discussed in section 2.4. By default, Boxer relies on C&C (Curran et al., 2009) to get the CCG parses of the sentences. Instead, we run Boxer on CCG parses produced by EasyCCG (Lewis and Steedman, 2014) because it is more accurate on FraCas. Like Lewis and Steedman (2013) we additionally test on gold-standard parses to be able to evaluate our technique of handling quantifiers in the absence of parser errors. Also, as we do with the SICK dataset, we add the inference $P(\neg T|H)$ for the detection of contradictions.

For multi-sentence examples, we add a simple co-reference resolution step that connects definite NPs across sentences. For example, *the right to live in Europe* in T1 and T3 should corefer in the following example:

T1: Every European has the right to live in Europe
T2: Every European is a person
T3: Every person who has the right to live in Europe can travel freely within Europe

We also added two rules encoding lexical knowledge, paraphrased as "a lot of $x \Rightarrow x$" and "one of $x \Rightarrow x$" to handle one of the examples, as lexical coverage is not the focus of our analysis.

Results The rightmost panel in Table 1 summarizes the results of our system for gold parses and system parses. We see that the Existence assumption is not needed in this dataset because it is constructed to test semantics and not presupposition. Results with Existence are lower because without Existence, three cases (the previous example is one of them) are correctly classified as Entail, but with Existence they are classified as Neutral. Without Existence the domain is empty, and $P(\neg T|H) = 0$ because $\neg T$, which is existentially quantified, is trivially false. With Existence added, $P(\neg T|H) = 1$ because the domain is not empty, and the CWA is not handled. Also we see that, as in the two previous experiments, adding the approach that handles the CWA has the biggest impact. With all components of the system added, and with gold parses, we get 100% accuracy. With system parses, all results are lower, but the relative scores for the different subsystems are comparable to the gold parse case.

5 Future Work

One important extension to this work is to support generalized quantifiers in probabilistic logic. Some determiners, such as "few" and "most", cannot be represented in standard first-order logic. But it could be possible to represent them using the probabilistic aspect of probabilistic logic. With support for generalized quantifiers, we would be able to evaluate on the pairs we skipped from the FraCas dataset (Cooper et al., 1996). Another important next step is to refine the closed world assumption such that it assumes fewer things to be false. In particular we want to use (hard or soft) typing to distinguish between impossible propositions on the one hand and possible but unsupported ones on the other hand.

[2] We use the version by MacCartney and Manning (2007)

[3] The first section consists of 80 pairs, but like MacCartney and Manning (2007) we ignore the pairs with an undefined result.

[4] The gold standard annotation for pair number 69 should be Neutral not Entail. We changed it accordingly.

6 Conclusion

In this paper we showed how to do textual entailment in probabilistic logic while taking into account the problems resulting from the domain closure assumption (DCA) and the closed-world assumption (CWA). This paper is not concerned with *why* one should use probabilistic logic for natural language semantics (we argue for the need for probabilistic logic in previous work (Beltagy et al., 2013)), it only addresses the *how*. Our formulation involves Skolemization of the text T, generating evidence for universally quantified variables in T to simulate the existential presupposition, generating evidence to test universal quantifiers in H, and preventing negated H from being judged true independently of T because of the CWA. We evaluate our formulation on a synthetic dataset that has complex forms of quantifications and on the relevant part of the FraCas dataset and get a 100% accuracy on both. We also evaluate on the SICK dataset and show improved performance.

Acknowledgments

This research was supported by the DARPA DEFT program under AFRL grant FA8750-13-2-0026 and by the NSF CAREER grant IIS 0845925. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the view of DARPA, DoD or the US government. Some experiments were run on the Mastodon Cluster supported by NSF Grant EIA-0303609. We are grateful to Raymond Mooney, as well as the anonymous reviewers, for helpful discussions.

References

Agirre, E., D. Cer, M. Diab, and A. Gonzalez-Agirre (2012). SemEval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval-2012)*.

Beltagy, I., C. Chau, G. Boleda, D. Garrette, K. Erk, and R. Mooney (2013). Montague meets Markov: Deep semantics with probabilistic logical form. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM-2013)*.

Beltagy, I. and R. J. Mooney (2014). Efficient Markov logic inference for natural language semantics. In *Proceedings of AAAI 2014 Workshop on Statistical Relational AI (StarAI-2014)*.

Berant, J., A. Chou, R. Frostig, and P. Liang (2013). Semantic parsing on Freebase from question-answer pairs. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2013)*.

Bos, J. (2008). Wide-coverage semantic analysis with Boxer. In *Proceedings of Semantics in Text Processing (STEP-2008)*.

Bos, J. (2013). Is there a place for logic in recognizing textual entailment? *Linguistic Issues in Language Technology 9*.

Choi, Y. and C. Cardie (2008). Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2008)*.

Cooper, R., D. Crouch, J. Van Eijck, C. Fox, J. Van Genabith, J. Jaspars, H. Kamp, D. Milward, M. Pinkal, M. Poesio, et al. (1996). Using the framework. Technical report, Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Copestake, A. and D. Flickinger (2000). An open-source grammar development environment and broad-coverage english grammar using HPSG. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC-2000)*.

Curran, J., S. Clark, and J. Bos (2009). Linguistically motivated large-scale nlp with c&c and boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*.

Dagan, I., D. Roth, M. Sammons, and F. M. Zanzotto (2013). Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies 6*(4), 1–220.

Ganitkevitch, J., B. Van Durme, and C. Callison-Burch (2013). PPDB: The paraphrase database. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies(NAACL-HLT-2013)*, pp. 758–764.

Garrette, D., K. Erk, and R. Mooney (2011). Integrating logical representations with probabilistic information using Markov logic. In *Proceedings of International Conference on Computational Semantics (IWCS-2011)*.

Genesereth, M. R. and N. J. Nilsson (1987). *Logical foundations of artificial intelligence*. San Mateo, CA: Morgan Kaufman.

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

Geurts, B. (2007). Existential import. In I. Comorovski and K. van Heusinger (Eds.), *Existence: syntax and semantics*, pp. 253–271. Dordrecht: Springer.

Kok, S., P. Singla, M. Richardson, and P. Domingos (2005). The Alchemy system for statistical relational AI. `http://www.cs.washington.edu/ai/alchemy`.

Kwiatkowski, T., E. Choi, Y. Artzi, and L. Zettlemoyer (2013). Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2013)*.

Lewis, M. and M. Steedman (2013). Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics (TACL-2013) 1*, 179–192.

Lewis, M. and M. Steedman (2014). A* ccg parsing with a supertag-factored model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2014)*.

Lin, D. and P. Pantel (2001). DIRT - discovery of inference rules from text. In *In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Lotan, A., A. Stern, and I. Dagan (2013). Truthteller: Annotating predicate truth. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies(NAACL-HLT-2013)*.

MacCartney, B. and C. D. Manning (2007). Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pp. 193–200.

MacCartney, B. and C. D. Manning (2008). Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling-2008)*.

MacCartney, B. and C. D. Manning (2009). An extended model of natural logic. In *Proceedings of the International Workshop on Computational Semantics (IWCS-2009)*.

Marelli, M., S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli (2014, may). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Montague, R. (1973). The proper treatment of quantification in ordinary English. In K. Hintikka, J. Moravcsik, and P. Suppes (Eds.), *Approaches to Natural Language*, pp. 221–242. Dordrecht: Reidel.

Nilsson, N. J. (1986). Probabilistic logic. *Artificial intelligence 28*(1), 71–87.

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

Skolem, T. (1920). Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit mathematischer Sätze. *Skrifter utgit av Videnskapselskapet i Kristiania 4*, 4–36.

Strawson, P. F. (1950). On referring. *Mind 59*, 320–344.