UTexas: Natural Language Semantics using Distributional Semantics and Probabilistic Logic

Islam Beltagy*, Stephen Roller*, Gemma Boleda†, Katrin Erk†, Raymond J. Mooney*
* Department of Computer Science
† Department of Linguistics
The University of Texas at Austin
{beltagy, roller, mooney}@cs.utexas.edu
gemma.boleda@upf.edu, katrin.erk@mail.utexas.edu

Abstract

We represent natural language semantics by combining logical and distributional information in probabilistic logic. We use Markov Logic Networks (MLN) for the RTE task, and Probabilistic Soft Logic (PSL) for the STS task. The system is evaluated on the SICK dataset. Our best system achieves 73% accuracy on the RTE task, and a Pearson's correlation of 0.71 on the STS task.

1 Introduction

Textual Entailment systems based on logical inference excel in correct reasoning, but are often brittle due to their inability to handle soft logical inferences. Systems based on distributional semantics excel in lexical and soft reasoning, but are unable to handle phenomena like negation and quantifiers. We present a system which takes the best of both approaches by combining distributional semantics with probabilistic logical inference.

Our system builds on our prior work (Beltagy et al., 2013; Beltagy et al., 2014a; Beltagy and Mooney, 2014; Beltagy et al., 2014b). We use Boxer (Bos, 2008), a wide-coverage semantic analysis tool to map natural sentences to logical form. Then, distributional information is encoded in the form of inference rules. We generate lexical and phrasal rules, and experiment with symmetric and asymmetric similarity measures. Finally, we use probabilistic logic frameworks to perform inference, Markov Logic Networks (MLN) for RTE, and Probabilistic Soft Logic (PSL) for STS.

2 Background

2.1 Logical Semantics

Logic-based representations of meaning have a long tradition (Montague, 1970; Kamp and Reyle, 1993). They handle many complex semantic phenomena such as relational propositions, logical operators, and quantifiers; however, they can not handle "graded" aspects of meaning in language because they are binary by nature.

2.2 Distributional Semantics

Distributional models use statistics of word co-occurrences to predict semantic similarity of words and phrases (Turney and Pantel, 2010; Mitchell and Lapata, 2010), based on the observation that semantically similar words occur in similar contexts. Words are represented as vectors in high dimensional spaces generated from their contexts. Also, it is possible to compute vector representations for larger phrases compositionally from their parts (Mitchell and Lapata, 2008; Mitchell and Lapata, 2010; Baroni and Zamparelli, 2010). Distributional similarity is usually a mixture of semantic relations, but particular *asymmetric* similarity measures can, to a certain extent, predict hypernymy and lexical entailment distributionally (Kotlerman et al., 2010; Lenci and Benotto, 2012; Roller et al., 2014). Distributional models capture the graded nature of meaning, but do not adequately capture logical structure (Grefenstette, 2013).

2.3 Markov Logic Network

Markov Logic Networks (MLN) (Richardson and Domingos, 2006) are a framework for probabilistic logic that employ weighted formulas in first-order logic to compactly encode complex undirected probabilistic graphical models (i.e., Markov networks). Weighting the rules is a way of softening them compared to hard logical constraints.

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 796–801,
Dublin, Ireland, August 23-24, 2014.

MLNs define a probability distribution over possible worlds, where the probability of a world increases exponentially with the total weight of the logical clauses that it satisfies. A variety of inference methods for MLNs have been developed, however, computational overhead is still an issue.

2.4 Probabilistic Soft Logic

Probabilistic Soft Logic (PSL) is another recently proposed framework for probabilistic logic (Kimmig et al., 2012). It uses logical representations to compactly define large graphical models with continuous variables, and includes methods for performing efficient probabilistic inference for the resulting models. A key distinguishing feature of PSL is that ground atoms (i.e., atoms without variables) have soft, continuous truth values on the interval [0, 1] rather than binary truth values as used in MLNs and most other probabilistic logics. Given a set of weighted inference rules, and with the help of Lukasiewicz's relaxation of the logical operators, PSL builds a graphical model defining a probability distribution over the continuous space of values of the random variables in the model (Kimmig et al., 2012). Then, PSL's MPE inference (Most Probable Explanation) finds the overall interpretation with the maximum probability given a set of evidence. This optimization problem is a second-order cone program (SOCP) (Kimmig et al., 2012) and can be solved in polynomial time.

2.5 Recognizing Textual Entailment

Recognizing Textual Entailment (RTE) is the task of determining whether one natural language text, the *premise*, Entails, Contradicts, or is not related (Neutral) to another, the *hypothesis*.

2.6 Semantic Textual Similarity

Semantic Textual Similarity (STS) is the task of judging the similarity of a pair of sentences on a scale from 1 to 5 (Agirre et al., 2012). Gold standard scores are averaged over multiple human annotations and systems are evaluated using the Pearson correlation between a system's output and gold standard scores.

3 Approach

3.1 Logical Representation

The first component in the system is Boxer (Bos, 2008), which maps the input sentences into logical form, in which the predicates are words in the sentence. For example, the sentence "A man is driving a car" in logical form is:

$\exists x, y, z.\ man(x) \land agent(y, x) \land drive(y) \land patient(y, z) \land car(z)$

3.2 Distributional Representation

Next, distributional information is encoded in the form of weighted inference rules connecting words and phrases of the input sentences T and H. For example, for sentences T: "A man is driving a car", and H: "A guy is driving a vehicle", we would like to generate rules like $\forall x.\ man(x) \Rightarrow guy(x) \,|\, w_1$, $\forall x. car(x) \Rightarrow vehicle(x) \,|\, w_2$, where w_1 and w_2 are weights indicating the similarity of the antecedent and consequent of each rule.

Inferences rules are generated as in Beltagy et al. (2013). Given two input sentences T and H, for all pairs (a, b), where a and b are words or phrases of T and H respectively, generate an inference rule: $a \rightarrow b \mid w$, where the rule weight w is a function of $sim(\overrightarrow{a}, \overrightarrow{b})$, and sim is a similarity measure of the distributional vectors $\overrightarrow{a}, \overrightarrow{b}$. We experimented with the symmetric similarity measure $cosine$, and $asym$, the supervised, asymmetric similarity measure of Roller et al. (2014).

The $asym$ measure uses the vector difference $(\overrightarrow{a} - \overrightarrow{b})$ as features in a logistic regression classifier for distinguishing between four different word relations: hypernymy, cohyponymy, meronomy, and no relation. The model is trained using the noun-noun subset of the BLESS data set (Baroni and Lenci, 2011). The final similarity weight is given by the model's estimated probability that the word relationship is either hypernymy or meronomy: $asym(\overrightarrow{a}, \overrightarrow{b}) = P(hyper(a, b)) + P(mero(a, b))$.

Distributional representations for words are derived by counting co-occurrences in the ukWaC, WaCkypedia, BNC and Gigaword corpora. We use the 2000 most frequent content words as basis dimensions, and count co-occurrences within a two word context window. The vector space is weighted using Positive Pointwise Mutual Information.

Phrases are defined in terms of Boxer's output to be more than one unary atom sharing the same variable like "a little kid" ($little(k) \land kid(k)$), or two unary atoms connected by a relation like "a man is driving" ($man(m) \land agent(d, m) \land drive(d)$). We compute vector representations of phrases using vector addition across the component predicates. We also tried computing phrase vectors using component-wise vector multiplication (Mitchell and Lapata, 2010), but found it performed marginally worse than addition.

3.3 Probabilistic Logical Inference

The last component is probabilistic logical inference. Given the logical form of the input sentences, and the weighted inference rules, we use them to build a probabilistic logic program whose solution is the answer to the target task. A probabilistic logic program consists of the evidence set E, the set of weighted first order logical expressions (rule base RB), and a query Q. Inference is the process of calculating $Pr(Q|E, RB)$.

3.4 Task 1: RTE using MLNs

MLNs are the probabilistic logic framework we use for the RTE task (we do not use PSL here as it shares the problems of fuzzy logic with probabilistic reasoning). The RTE classification problem for the relation between T and H can be split into two inference tasks. The first is testing if T entails H, $Pr(H|T, RB)$. The second is testing if the negation of the text $\neg T$ entails H, $Pr(H|\neg T, RB)$. In case $Pr(H|T, RB)$ is high, while $Pr(H|\neg T, RB)$ is low, this indicates Entails. In case it is the other way around, this indicates Contradicts. If both values are close, this means T does not affect the probability of H and indicative of Neutral. We train an SVM classifier with LibSVM's default parameters to map the two probabilities to the final decision.

The MLN implementation we use is Alchemy (Kok et al., 2005). Queries in Alchemy can only be ground atoms. However, in our case the query is a complex formula (H). We extended Alchemy to calculate probabilities of queries (Beltagy and Mooney, 2014). Probability of a formula Q given an MLN K equals the ratio between the partition function Z of the ground network of K with and without Q added as a hard rule (Gogate and Domingos, 2011)

$$P(Q \mid K) = \frac{Z(K \cup \{(Q, \infty)\})}{Z(K)} \qquad (1)$$

We estimate Z of the ground networks using SampleSearch (Gogate and Dechter, 2011), an advanced importance sampling algorithm that is suitable for ground networks generated by MLNs.

A general problem with MLN inference is its computational overhead, especially for the complex logical formulae generated by our approach. To make inference faster, we reduce the size of the ground network through an automatic type-checking technique proposed in Beltagy and Mooney (2014). For example, consider the evidence ground atom $man(M)$ denoting that the constant M is of type man. Then, consider another predicate like $car(x)$. In case there are no inference rule connecting $man(x)$ and $car(x)$, then we know that M which we know is a man cannot be a car, so we remove the ground atom $car(M)$ from the ground network. This technique reduces the size of the ground network dramatically and makes inference tractable.

Another problem with MLN inference is that quantifiers sometimes behave in an undesirable way, due to the Domain Closure Assumption (Richardson and Domingos, 2006) that MLNs make. For example, consider the text-hypothesis pair: “There is a black bird” and “All birds are black”, which in logic are $T : bird(B) \wedge black(B)$ and $H : \forall x.\ bird(x) \Rightarrow black(x)$. Because of the Domain Closure Assumption, MLNs conclude that T entails H because H is true for all constants in the domain (in this example, the single constant B). We solve this problem by introducing extra constants and evidence in the domain. In the example above, we introduce evidence of a new bird $bird(D)$, which prevents the hypothesis from being true. The full details of the technique of dealing with the domain closure is beyond the scope of this paper.

3.5 Task 2: STS using PSL

PSL is the probabilistic logic we use for the STS task since it has been shown to be an effective approach for computing similarity between structured objects. We showed in Beltagy et al. (2014a) how to perform the STS task using PSL. PSL does not work “out of the box” for STS, because Lukasiewicz's equation for the conjunction is very restrictive. We address this by replacing Lukasiewicz's equation for conjunction with an averaging equation, then change the optimization problem and grounding technique accordingly.

For each STS pair of sentences S_1, S_2, we run PSL twice, once where $E = S_1$, $Q = S_2$ and another where $E = S_2$, $Q = S_1$, and output the two scores. The final similarity score is produced from an Additive Regression model with WEKA's default parameters trained to map the two PSL scores to the overall similarity score (Friedman, 1999; Hall et al., 2009).

3.6 Task 3: RTE and STS using Vector Spaces and Keyword Counts

As a baseline, we also attempt both the RTE and STS tasks using only vector representations and unigram counts. This baseline model uses a supervised regressor with features based on vector similarity and keyword counts. The same input features are used for performing RTE and STS, but a SVM classifier and Additive Regression model is trained separately for each task. This baseline is meant to establish whether the task truly requires the sophisticated logical inference of MLNs and PSL, or if merely checking for logical keywords and textual similarity is sufficient.

The first two features are simply the $cosine$ and $asym$ similarities between the text and hypothesis, using vector addition of the unigrams to compute a single vector for the entire sentence.

We also compute vectors for both the text and hypothesis using vector addition of the mutually exclusive unigrams (MEUs). The MEUs are defined as the unigrams of the premise and hypothesis with common unigrams removed. For example, if the premise is "A dog chased a cat" and the hypothesis is "A dog watched a mouse", the MEUs are "chased cat" and "watched mouse." We compute vector addition of the MEUs, and compute similarity using both the $cosine$ and $asym$ measures. These form two features for the regressor.

The last feature of the model is a keyword count. We count how many times 13 different keywords appear in either the text or the hypothesis. These keywords include negation (*no*, *not*, *nobody*, etc.) and quantifiers (*a*, *the*, *some*, etc.) The counts of each keyword form the last 13 features as input to the regressor. In total, there are 17 features used in this baseline system.

4 Evaluation

The dataset used for evaluation is **SICK**: Sentences Involving Compositional Knowledge dataset, a task for SemEval 2014 (Marelli et al., 2014a; Marelli et al., 2014b). The dataset is 10,000 pairs of sentences, 5000 training and 5000 for testing. Sentences are annotated for both tasks.

	SICK-RTE	SICK-STS
Baseline	70.0	71.1
MLN/PSL + Cosine	72.8	68.6
MLN/PSL + Asym	73.2	68.9
Ensemble	73.2	71.5

Table 1: Test RTE accuracy and STS Correlation.

4.1 Systems Compared

We compare multiple configurations of our probabilistic logic system.

- **Baseline**: Vector- and keyword-only baseline described in Section 3.6;
- **MLN/PSL + Cosine**: MLN and PSL based methods described in Sections 3.4 and 3.5, using $cosine$ as a similarity measure;
- **MLN/PSL + Asym**: MLN and PSL based methods described in Sections 3.4 and 3.5, using $asym$ as a similarity measure;
- **Ensemble**: An ensemble method which uses all of the features in the above methods as inputs for the RTE and STS classifiers.

4.2 Results and Discussion

Table 1 shows our results on the held-out test set for SemEval 2014 Task 1.

On the RTE task, we see that both the MLN + Cosine and MLN + Asym models outperformed the Baseline, indicating that textual entailment requires real inference to handle negation and quantifiers. The MLN + Asym and Ensemble systems perform identically on RTE, further suggesting that the logical inference subsumes keyword detection.

The MLN + Asym system outperforms the MLN + Cosine system, emphasizing the importance of asymmetric measures for predicting lexical entailment. Intuitively, this makes perfect sense: *dog* entails *animal*, but not vice versa.

In an error analysis performed on a development set, we found our RTE system was extremely conservative: we rarely confused the Entails and Contradicts classes, indicating we correctly predict the direction of entailment, but frequently misclassify examples as Neutral. An examination of these examples showed the errors were mostly due to missing or weakly-weighted distributional rules.

On STS, our vector space baseline outperforms both PSL-based systems, but the ensemble outperforms any of its components. This is a testament to the power of distributional models in their ability to predict word and sentence similarity. Surprisingly, we see that the PSL + Asym system slightly outperforms the PSL + Cosine system. This may indicate that even in STS, some notion of asymmetry plays a role, or that annotators may have been biased by simultaneously annotating both tasks. As with RTE, the major bottleneck of our system appears to be the knowledge base, which is built solely using distributional inference rules.

Results also show that our system's performance is close to the baseline system. One of the reasons behind that could be that sentences are not exploiting the full power of logical representations. On RTE for example, most of the contradicting pairs are two similar sentences with one of them being negated. This way, the existence of any negation cue in one of the two sentences is a strong signal for contradiction, which what the baseline system does without deeply representing the semantics of the negation.

5 Conclusion & Future Work

We showed how to combine logical and distributional semantics using probabilistic logic, and how to perform the RTE and STS tasks using it. The system is tested on the SICK dataset.

The distributional side can be extended in many directions. We would like to use longer phrases, more sophisticated compositionality techniques, and contextualized vectors of word meaning. We also believe inference rules could be dramatically improved by integrating from paraphrases collections like PPDB (Ganitkevitch et al., 2013).

Finally, MLN inference could be made more efficient by exploiting the similarities between the two ground networks (the one with Q and the one without). PLS inference could be enhanced by using a learned, weighted average of rules, rather than the simple mean.

Acknowledgements

This research was supported by the DARPA DEFT program under AFRL grant FA8750-13-2-0026. Some experiments were run on the Mastodon Cluster supported by NSF Grant EIA-0303609. The authors acknowledge the Texas Advanced Computing Center (TACC)[1] for providing grid resources that have contributed to these results. We thank the anonymous reviewers and the UTexas Natural Language and Learning group for their helpful comments and suggestions.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of Semantic Evaluation (SemEval-12)*.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK, July. Association for Computational Linguistics.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-10)*.

Islam Beltagy and Raymond J. Mooney. 2014. Efficient Markov logic inference for natural language semantics. In *Proceedings of AAAI 2014 Workshop on Statistical Relational AI (StarAI-14)*.

Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond Mooney. 2013. Montague meets Markov: Deep semantics with probabilistic logical form. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM-13)*.

Islam Beltagy, Katrin Erk, and Raymond Mooney. 2014a. Probabilistic soft logic for semantic textual similarity. In *Proceedings of Association for Computational Linguistics (ACL-14)*.

Islam Beltagy, Katrin Erk, and Raymond Mooney. 2014b. Semantic parsing using distributional semantics and probabilistic logic. In *Proceedings of ACL 2014 Workshop on Semantic Parsing (SP-2014)*.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Proceedings of Semantics in Text Processing (STEP-08)*.

J.H. Friedman. 1999. Stochastic gradient boosting. Technical report, Stanford University.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-13)*.

Vibhav Gogate and Rina Dechter. 2011. Samplesearch: Importance sampling in presence of determinism. *Artificial Intelligence*, 175(2):694–729.

Vibhav Gogate and Pedro Domingos. 2011. Probabilistic theorem proving. In *27th Conference on Uncertainty in Artificial Intelligence (UAI-11)*.

[1] http://www.tacc.utexas.edu

Edward Grefenstette. 2013. Towards a formal distributional semantics: Simulating logical calculi with tensors. In *Proceedings of Second Joint Conference on Lexical and Computational Semantics (*SEM 2013)*.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic*. Kluwer.

Angelika Kimmig, Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2012. A short introduction to Probabilistic Soft Logic. In *Proceedings of NIPS Workshop on Probabilistic Programming: Foundations and Applications (NIPS Workshop-12)*.

Stanley Kok, Parag Singla, Matthew Richardson, and Pedro Domingos. 2005. The Alchemy system for statistical relational AI. Technical report, Department of Computer Science and Engineering, University of Washington. `http://www.cs.washington.edu/ai/alchemy`.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In *Proceedings of the first Joint Conference on Lexical and Computational Semantics (*SEM-12)*.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014a. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014b. A sick cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of Association for Computational Linguistics (ACL-08)*.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(3):1388–1429.

Richard Montague. 1970. Universal grammar. *Theoria*, 36:373–398.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62:107–136.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of the Twenty Fifth International Conference on Computational Linguistics (COLING-14)*, Dublin, Ireland.

Peter Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.