

Instances and Concepts in Distributional Space

Gemma Boleda

Universitat Pompeu Fabra
Barcelona, Spain

gemma.boleda@upf.edu

Abhijeet Gupta and Sebastian Padó

Universität Stuttgart
Stuttgart, Germany

guptaat,pado@ims.uni-stuttgart.de

Abstract

Instances (“Mozart”) are ontologically distinct from concepts or classes (“composer”). Natural language encompasses both, but instances have received comparatively little attention in distributional semantics. Our results show that instances and concepts differ in their distributional properties. We also establish that instantiation detection (“Mozart – composer”) is generally easier than hypernymy detection (“chemist – scientist”), and that results on the influence of input representation do not transfer from hyponymy to instantiation.

1 Introduction

Distributional semantics (Turney and Pantel, 2010), and data-driven, continuous approaches to language in general including neural networks (Bengio et al., 2003), are a success story in both Computational Linguistics and Cognitive Science in terms of modeling *conceptual* knowledge, such as the fact that cats are animals (Baroni et al., 2012), similar to dogs (Landauer and Dumais, 1997), and shed fur (Erk et al., 2010). However, distributional representations are notoriously bad at handling discrete knowledge (Fodor and Lepore, 1999; Smolensky, 1990), such as information about specific instances. For example, Beltagy et al. (2016) had to revert from a distributional to a symbolic knowledge source in an entailment task because the distributional component licensed unwarranted inferences (*white man* does not entail *black man*, even though the phrases are distributionally very similar). This partially explains that instances have received much less attention than concepts in distributional semantics.

This paper addresses this gap and shows that distributional models can reproduce the age-old

ontological distinction between instances and concepts. Our work is exploratory: We seek insights into how distributional representations mirror the instance/concept distinction and the hypernymy/instantiation relations.

Our contributions are as follows. First, we build publicly available datasets for instantiation and hypernymy (Section 2).¹ Second, we carry out a contrastive analysis of instances and concepts, finding substantial differences in their distributional behavior (Section 3). Finally, in Section 4, we compare supervised models for instantiation detection (*Lincoln – president*) with such models for hypernymy detection (*19th century president – president*). Identifying instantiation turns out to be easier than identifying hypernymy in our experiments.

2 Datasets

We focus on “public” named entities such as Abraham Lincoln or Vancouver, as opposed to “private” named entities like my neighbor Michael Smith or unnamed entities like the bird I saw today), because for public entities we can extract distributional representations directly from corpus data.²

No existing dataset treats entities and concepts on a par, which would enable a contrastive analysis of instances and concepts. Therefore, we create the data for our study, building two comparable datasets around the binary semantic relations of *instantiation* and *hypernymy* (see Table 2). This design enables us to relate our results to work on hypernymy (see Section 5), and provides a rich relational perspective on the instance–concept divide: In both cases, we are dealing with the relationship

¹Available from <http://www.ims.uni-stuttgart.de/data/Instantiation.html>.

²Note that, for feasibility reasons, our distributional representations are made up of explicit mentions of proper nouns (*Abraham Lincoln, Lincoln*), without taking into account other referential expressions (*he, the 16th president of the United States, the president*). We leave these to future work.

	INSTANCE	HYPERNYM
Total	28,424	30,488
Positive	7,106	7,622
Unique inst./hypo.	5,847	7,622
Unique conc./hyper.	540	2,369

Table 1: Dataset statistics. *Total* number of datapoints, *Positive* cases, unique instances/hyponyms and unique concepts/hypernyms.

	INSTANCE	HYPERNYM
Positive	<i>Mozart – composer</i>	<i>chemist – scientist</i>
NOTINST/ NOTHYP	<i>Mozart – garden</i>	<i>chemist – communication</i>
INVERSE	<i>composer – Mozart</i>	<i>scientist – chemist</i>
I2I/C2C	<i>Mozart – O. Robertson</i>	<i>chemist – diadem</i>

Table 2: Positive examples and confounders.

between a more general (concept/hypernym) and a more specific object (instance/hyponym), but, from an ontological perspective, hyponym concepts, as classes of individuals, are considered to be completely different from instances, both in theoretical linguistics and in AI (Dowty et al., 1981; Lenat and Guha, 1990; Fellbaum, 1998).

We construct both datasets from the WordNet noun hierarchy. Its backbone is formed by hyponymy (Fellbaum, 1998) and it was later extended with instance-concept links marked with the `Hypernym_Instance` relation (Miller and Hristea, 2006). We sample the items from WordNet that are included in the space we will use in the experiments, namely, the word2vec entity vector space, which is, to our knowledge, the largest existing source for entity vectors.³ The space was trained on Google News, and contains vectors for nodes in FreeBase which covers millions of entities and thousands of concepts. This enables us to perform comparative analyses, as we sample instances and concepts from a common resource, and that we have compatible vector representations for both.

INSTANCE. This dataset contains around 30K datapoints for instantiation (see Table 1 for statistics and Table 2 for examples).⁴ It contains 7K positive cases (e.g., *Vancouver-city*), namely all pairs of instances and their concepts from WordNet that are covered by the word2vec entity vector

³<https://code.google.com/p/word2vec>

⁴Each instance can belong to multiple concepts (*Vancouver-city* and *Vancouver-port*), and different instances/hyponyms can belong to the same concept/hypernym.

	Global sim.	Local sim.
Instances	0.045 (0.02)	0.528 (0.16)
Concepts	0.037 (0.02)	0.390 (0.12)
Instance-Concept	0.021 (0.01)	0.379 (0.12)

Table 3: Cosine similarities for within-type and across-type pairs (means and standard deviations).

space. For each positive example, we create three confounders, or negative examples, as follows:

1. The NOTINST subset pairs the instance with a wrong concept, to ensure that we do not only spot instances vs. concepts in general, but truly detect the instantiation relationship.
2. The INVERSE subset switches instance and concept, to check that we are capturing the asymmetry in the relationship.
3. The I2I (instance-to-instance) subset pairs the instance with a random instance from another concept, a sanity check to ensure that we are not thrown off by the high similarity among instances (see Section 3).

HYPERNYMY. This dataset contains hypernymy examples which are as similar to the INSTANCE dataset as possible. The set of potential hyponyms are obtained from the intersection between the nouns in the word2vec entity space and WordNet, excluding instances. Each of the nouns that has a direct WordNet hypernym as well as a co-hyponym is combined with the direct hypernym into a positive example. The confounders are then built in parallel to those for INSTANCE. Note that in this case the equivalent of NOTINST is actually not-hypernym (hence NOTHYP in the results discussion), and the equivalent of I2I is concept-to-concept (C2C).⁵

3 Instances and Concepts

We first explore the differences between instances and concepts by comparing the *distribution of similarities* of their word2vec vectors (cf. previous section). We use both a global measure of similarity (average cosine to all other members of the respective set), and a local measure (cosine to the nearest neighbor). The results, shown in Table 3, indicate that instances exhibit substantially higher similarities than concepts, both at the global and at

⁵This does not reduce to co-hyponymy, because the hyponym is randomly paired with another hyponym.

the local level.⁶ The difference holds even though we consider more unique concepts than instances (Table 1), and might thus expect the concepts to show higher similarities, at least at the local level. The global similarity of instances and concepts is the lowest (see last row in Table 3), suggesting that instances and concepts are represented distinctly in the space, even when they come from the same domain (here, newswire).

Taken together, these observations indicate that instances are *semantically more coherent* than concepts, at least in our space. We believe a crucial reason for this is that instances share the same specificity, referring to one entity, while concepts are of widely varying specificity and size (compare *president of the United States* with *artifact*). Further work is required to probe this hypothesis.

It is well established in lexical semantics that cosine similarity does not distinguish between hyponymy and other lexical relations, and in fact hyponyms and hypernyms are usually less similar than co-hyponyms like *cat-dog* or antonyms like *good-bad* (Baroni and Lenci, 2011). This result extends to instantiation: The average similarity of each instance to its concept is 0.110 (standard deviation: 0.12), very low compared to the figures in Table 3. The nearest neighbors of instances show a wide range of relations similar to those of concepts, further enriched by the instance-concept axis: *Tyre - Syria* (location), *Thames river - estuary* (“co-hyponym class”), *Luciano Pavarotti - soprano* (“contrastive class”), *Joseph Goebels - bolshevik* (“antonym class”), and occasionally true instantiation cases like *Sidney Poitier - actor*.

4 Modeling Instantiation vs. Hypernymy

The analysis in the previous section suggests clearly that unsupervised methods are not adequate for instantiation, so we turn to supervised methods, which have also been used for hypernymy detection (Baroni et al., 2012; Roller et al., 2014). Also note that unsupervised asymmetric measures previously used for hypernymy (Lenci and Benotto, 2012; Santus et al., 2014) are only applicable to non-negative vector spaces, which excludes predictive models like the one we use.

We use a logistic regression classifier, partitioning the data into train/dev/test portions (80/10/10%) and ensuring that instances/hyponyms are not

⁶Both differences are statistically significant at $\alpha=0.001$ according to a Kruskal-Wallis test.

reused across partitions. We report F-scores for the positive class on the test sets.

Table 4 shows the results. Rows correspond to experiments. The task is always to detect instantiation (left) or hypernymy (right), but the confounders differ: We combine the positive examples with each of the individual negative datasets (NOTINST/NOTHYP, INVERSE, I2I/C2C, cf. Section 2, all balanced setups) and with the union of all negative datasets (UNION, 25% positive examples). The columns correspond to feature sets. We consider two baselines: *Freq* for most frequent class, *IVec* for a baseline where the classifier only sees the vector for the first component of the input pair – for instance, for NOTINST, only the instance vector is given. This baseline tests possible memorization effects (Levy et al., 2015). For instantiation, we have a third baseline, *Cap*. It makes a rule-based decision on the basis of capitalization where available and guesses randomly otherwise. The remaining columns show results for three representations that have worked well for hypernymy (see Roller et al. (2014) and below for discussion): Concatenating the two input vectors (*Conc*), their difference (*Diff*), and concatenating the difference vector with the squared difference vector (*DDSq*).

Instantiation. Instantiation achieves overall quite good results, well above the baselines and with nearly perfect F-score for the INVERSE and I2I cases. Recall that these setups basically require the classifier to characterize the notion of instance vs. concept, which turns out to be an easy task, consistent with the analysis in the previous section. Indeed, for INVERSE, the *IVec* and *Cap* baselines also achieve (near-)perfect F-scores of 0.96 and 1.00 respectively; in this case, the input is either an instance or a concept vector, so the task reduces to instance identification. The distributional models perform at the same level (0.98-0.99).

The most difficult setup is NOTINST, where the model has to decide whether the concept matches the instance, with 0.79 best performance. Since the INVERSE and I2I cases are easy, the combined task is about as difficult as NOTINST, and the best result for UNION is the same (0.79). The very bad performance of *IVec* in this case excludes memorization as a significant factor in our setup.

Instantiation vs. Hypernymy. Table 4 shows that, in our setup, hypernymy detection is considerably harder than instantiation: Results are 0.57-

INSTANCE	Freq	1Vec	Cap	Conc	Diff	DDSq	HYPERNYM	Freq	1Vec	Conc	Diff	DDSq
NOTINST	0.49	0.32	0.67	0.79	0.77	0.78	NOTHYP	0.51	0.29	0.55	0.53	0.57
INVERSE	0.5	0.96	1.00	0.98	0.99	0.99	INVERSE	0.5	0.65	0.75	0.78	0.78
I2I	0.5	0.31	0.80	0.97	0.94	0.94	C2C	0.51	0.29	0.64	0.58	0.62
UNION	0.25	0.01	0.57	0.79	0.74	0.74	UNION	0.25	0.00	0.31	0.26	0.30

Table 4: Supervised modeling results (rows: datasets/tasks, columns: feature sets)

0.78 for the individual hypernymy tasks, compared to the 0.79-0.99 range of instantiation.⁷ The difference is even more striking for UNION, with 0.31 vs. 0.79. Our interpretation is that, in contrast to instantiation, the individual tasks for hypernymy are all nontrivial, such that modeling them together is substantially more difficult. INVERSE and C2C require the classifier to model the notion of concept specificity (other concepts may be semantically related, but what distinguishes hypernymy is the fact that hyponyms are more specific), which is apparently more difficult than characterizing the notion of instance as opposed to concept.

Frequency Effects. We now test the effect of frequency on our best model (Conc) on the most interesting dataset family (UNION). The word2vec vectors do not provide absolute frequencies, but frequency ranks. Thus, we rank-order our two datasets, split each into ten deciles, and compute new F-Scores. The results in Figure 1 show that there are only mild effects of frequency, in particular compared to the general level of inter-bin variance: for INSTANCE, the lowest-frequency decile yields an F-Score of 76% compared to 81% for the highest-frequency one. The numbers are comparable for the HYPERNYM dataset, with 28% and 36%, respectively. We conclude that frequency is not a decisive factor in our present setup.

Input Representation. Regarding the effect of the input representation, we reproduce Roller et al.’s (2014) results that DDSq works best for hypernymy detection in the NOTHYP setup. In contrast, for instantiation detection it is the concatenation of the input vectors that works best (cf. NOTINST row in Table 4). Difference features (*Diff*, *DDSq*) perform a pre-feature selection, signaling systematic commonalities and differences in distributional representations as well as the direction of feature in-

clusion; Roller et al. (2014) argued that the squared difference features “identify dimensions that are not indicative of hypernymy”, thus removing noise. Concatenating vectors, instead, allows the classifier to combine the information in the features more freely. We thus take our results to suggest that the relationship between instances and their concept is overall *less predictable* than the relationship between hyponyms and hypernyms. This appears plausible given the tendency of instances to be more “crisp”, or idiosyncratic, in their properties than concepts (compare the relation between Mozart or John Lennon and *composer* with that of *poet* or *novelist* and *writer*). This interpretation is also consistent with the fact that difference features work best for the INVERSE case, which requires characterizing the notion of inclusion, and concatenation works best for the I2I and C2C cases, where instead we are handling potentially unrelated instances or concepts.

Error analysis. An error analysis on the most interesting INSTANCE setup (UNION dataset with *Conc* features) reveals errors typical for distributional approaches. The first major error source is ambiguity. For example, WordNet often lists multiple “senses” for named entities (*Washington* as synonym for *George Washington* and a city name, a.o.). The corresponding vector representations are mixtures of the contexts of the individual entities and consequently more difficult to process, no matter which sense we consider. The second major error source is general semantic relatedness. For instance, the model predicts that the writer *Franz Kafka* is a *Statesman*, presumably due to the bureaucratic topics of his novels that are often discussed in connection with his name. Similarly, *Arnold Schönberg* – *writer* is due to Schönberg’s work as a music theorist. Finally, *Einstein* – *river* combines both error types: Hans A. Einstein, Albert Einstein’s son, was an expert on sedimentation.

5 Related Work

Recent work has started exploring the representation of instances in distributional space: Herbe-

⁷Our hypernymy results are lower than previous work. E.g. Roller et al. (2014) report 0.85 maximum accuracy on a task analogous to NOTHYP, compared to our 0.57 F-score. Since our results are not directly comparable in terms of evaluation metric, dataset, and space, we leave it to future work to examine the influence of these factors.

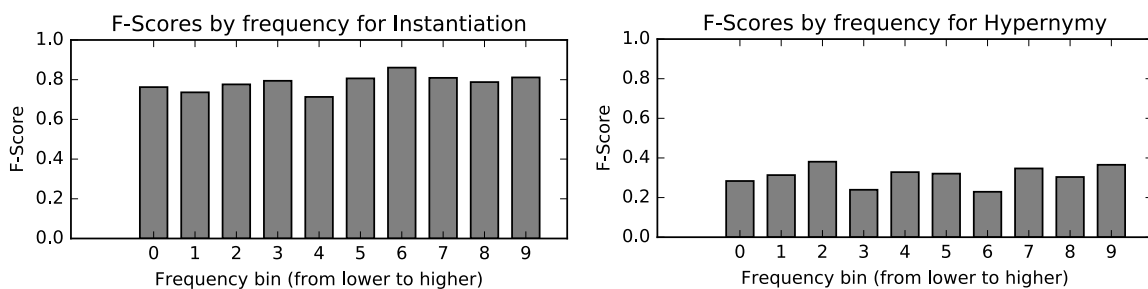


Figure 1: Performance by frequency bin

lot and Vecchi (2015) and Gupta et al. (2015) extract quantified and specific properties of instances (some cats are black, Germany has 80 million inhabitants), and Kruszewski et al. (2015) seek to derive a semantic space where dimensions are sets of entities. We instead analyze instance vectors. A similar angle is taken in Herbelot and Vecchi (2015), for “artificial” entity vectors, whereas we explore “real” instance vectors extracted with standard distributional methods. An early exploration of the properties of instances and concepts, limited to a few manually defined features, is Alfonseca and Manandhar (2002).

Some previous work uses distributional representations of instances for NLP tasks: For instance, Lewis and Steedman (2013) use the distributional similarity of named entities to build a type system for a semantic parser, and several works in Knowledge Base completion use entity embeddings (see Wang et al. (2014) and references there).

The focus on public, named instances is shared with Named Entity Recognition (NER; see Lample et al. (2016) and references therein); however, we focus on the instantiation relation rather than on recognition *per se*. Also, in terms of modeling, NER is typically framed as a sequence labeling task to identify entities in text, whereas we do classification of previously gathered candidates. In fact, the space we used was built on top of a corpus processed with a NER system. Named Entity Classification (Nadeau and Sekine, 2007) can be viewed as a limited form of the instantiation task. We analyze the entity representations themselves and tackle a wider set of tasks related to instantiation, with a comparative analysis with hypernymy.


There is a large body of work on hypernymy and other lexical relations in distributional semantics (Geffet and Dagan, 2005; Kotlerman et al., 2010; Baroni and Lenci, 2011; Lenci and Benotto, 2012; Weeds et al., 2014; Rimell, 2014; Roller et

al., 2014; Santus et al., 2014; Levy et al., 2015; Santus et al., 2016; Roller and Erk, 2016; Shwartz et al., 2016). Many studies, notably studies of textual entailment, include entities, but do not specifically investigate their properties and contrast them with concepts: This is the contribution of our paper.

6 Conclusions

The ontological distinction between instances and concepts is fundamental both in theoretical studies and practical implementations. Our analyses and experiments suggest that the distinction is recoverable from distributional representations. The good news is that instantiation is easier to spot than hypernymy, consistent with it lying along a greater ontological divide. The bad (though expected) news is that not all extant results for concepts carry over to instances, for instance regarding input representation in classification tasks.

More work is required to better assess the properties of instances as well as the effects of design factors such as the underlying space and dataset construction. An extremely interesting (and challenging) extension is to tackle “anonymous” entities for which standard distributional techniques do not work (my neighbor, the bird we saw this morning), in the spirit of Herbelot and Vecchi (2015) and Boleda et al. (2017).

Acknowledgments. The authors have received funding from DFG (SFB 732, project B9). and from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 715154; AMORE), as well as under the Marie Skłodowska-Curie grant agreement No 655577 (LOVe). This paper reflects the authors’ view only, and the EU is not responsible for any use that may be made of the information it contains. 

References

- Enrique Alfonseca and Suresh Manandhar. 2002. Distinguishing concepts and instances in WordNet. In *Proceedings of the First International Conference of Global WordNet Association*, Mysore, India.
- 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, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France, April. Association for Computational Linguistics.
- Islam Beltagy, Stephen Roller, Pengiang Cheng, Katrin Erk, and Raymond Mooney. 2016. Representing Meaning with a Combination of Logical and Distributional Models. *Computational Linguistics*, 42(4).
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.
- Gemma Boleda, Sebastian Padó, Nghia The Pham, and Marco Baroni. 2017. Living a discrete life in a continuous world: Reference with distributed representations. *ArXiv e-prints*, February.
- David Dowty, Robert Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*. Riedel, Dordrecht.
- Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763.
- Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA; London.
- Jerry Fodor and Ernie Lepore. 1999. All at Sea in Semantic Space: Churchland on Meaning Similarity. *Journal of Philosophy*, 96(8):381–403.
- Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*, pages 107–114, Ann Arbor, Michigan, June. Association for Computational Linguistics.
- Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal, September. Association for Computational Linguistics.
- Aurélie Herbelot and Eva Maria Vecchi. 2015. Building a shared world: mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Lisbon, Portugal, September. Association for Computational Linguistics.
- Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.
- Germán Kruszewski, Denis Paperno, and Marco Baroni. 2015. Deriving Boolean structures from distributional vectors. *Transactions of the Association for Computational Linguistics*, 3:375–388.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June. Association for Computational Linguistics.
- Thomas K. Landauer and Susan T. Dumais. 1997. A Solution to Plato’s Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological review*, 104(2):211–240.
- Doug B. Lenat and Ramanathan V. Guha. 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley.
- Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79, Montréal, Canada, 7–8 June. Association for Computational Linguistics.
- Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado, May–June. Association for Computational Linguistics.
- Mike Lewis and Mark Steedman. 2013. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

- George A. Miller and Florentina Hristea. 2006. Word-net nouns: Classes and instances. *Computational Linguistics*, 32(1):1–3, 2016/12/15.
- David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26.
- Laura Rimell. 2014. Distributional lexical entailment by topic coherence. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 511–519, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Stephen Roller and Katrin Erk. 2016. Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2163–2172, Austin, Texas, November. Association for Computational Linguistics.
- Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.
- Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42, Gothenburg, Sweden, April. Association for Computational Linguistics.
- Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016. What a nerd! Beating students and vector cosine in the ESL and TOEFL datasets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany, August. Association for Computational Linguistics.
- Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1-2):159–216.
- Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1591–1601, Doha, Qatar, October. Association for Computational Linguistics.
- Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.