Entailment above the word level in distributional semantics

Marco Baroni
Raffaella Bernardi
University of Trento
name.surname@unitn.it

Ngoc-Quynh Do
Free University of Bozen-Bolzano
quynhdtn.hut@gmail.com

Chung-chieh Shan
Cornell University
University of Tsukuba
ccshan@post.harvard.edu

Abstract

We introduce two ways to detect entailment using distributional semantic representations of phrases. Our first experiment shows that the entailment relation between adjective-noun constructions and their head nouns (*big cat* $\models$ *cat*), once represented as semantic vector pairs, generalizes to lexical entailment among nouns (*dog* $\models$ *animal*). Our second experiment shows that a classifier fed semantic vector pairs can similarly generalize the entailment relation among quantifier phrases (*many dogs*$\models$*some dogs*) to entailment involving unseen quantifiers (*all cats*$\models$*several cats*). Moreover, nominal and quantifier phrase entailment appears to be cued by different distributional correlates, as predicted by the type-based view of entailment in formal semantics.

1 Introduction

Distributional semantics (DS) approximates linguistic meaning with vectors summarizing the contexts where expressions occur. The success of DS in lexical semantics has validated the hypothesis that semantically similar expressions occur in similar contexts (Landauer and Dumais, 1997; Lund and Burgess, 1996; Sahlgren, 2006; Schütze, 1997; Turney and Pantel, 2010). Formal semantics (FS) represents linguistic meanings as symbolic formulas and assemble them via composition rules. FS has successfully modeled quantification and captured inferential relations between phrases and between sentences (Montague, 1970; Thomason, 1974; Heim and Kratzer, 1998). The strengths of DS and FS have been complementary to date: On one hand, DS has induced large-scale semantic representations from corpora, but it has been largely limited to the lexical domain. On the other hand, FS has provided sophisticated models of sentence meaning, but it has been largely limited to hand-coded models that do not scale up to real-life challenges by learning from data.

Given these complementary strengths, we naturally ask if DS and FS can address each other's limitations. Two recent strands of research are bringing DS closer to meeting core FS challenges. One strand attempts to model compositionality with DS methods, representing both primitive and composed linguistic expressions as distributional vectors (Baroni and Zamparelli, 2010; Grefenstette and Sadrzadeh, 2011; Guevara, 2010; Mitchell and Lapata, 2010). The other strand attempts to reformulate FS's notion of logical inference in terms that DS can capture (Erk, 2009; Geffet and Dagan, 2005; Kotlerman et al., 2010; Zhitomirsky-Geffet and Dagan, 2010). In keeping with the lexical emphasis of DS, this strand has focused on inference at the word level, or *lexical entailment*, that is, discovering from distributional vectors of hyponyms (*dog*) that they entail their hypernyms (*animal*).

This paper brings these two strands of research together by demonstrating two ways in which the distributional vectors of composite expressions bear on inference. Here we focus on phrasal vectors harvested directly from the corpus rather than obtained compositionally. In a first experiment, we exploit the entailment properties of a class of composite expressions, namely adjective-noun constructions (ANs), to harvest training data for an entailment recognizer. The recognizer is then successfully applied to detect lexical entailment. In short, since almost all ANs entail the noun they contain (*red car* entails *car*), the distributional vectors of AN-N pairs can train a classifier to detect noun pairs that stand in the same relation (*dog*

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 23–32,
Avignon, France, April 23 - 27 2012.

entails *animal*). With almost no manual effort, we achieve performance nearly identical with the state-of-the-art balAPinc measure that Kotlerman et al. (2010) crafted, which detects feature inclusion between the two nouns' occurrence contexts.

Our second experiment goes beyond lexical inference. We look at phrases built from a quantifying determiner[1] and a noun (QNs) and use their distributional vectors to recognize entailment relations of the form *many dogs* $\models$ *some dogs*, between two QNs sharing the same noun. It turns out that a classifier trained on a set of $Q_1N \models Q_2N$ pairs can recognize entailment in pairs with a new quantifier configuration. For example, we can train on *many dogs* $\models$ *some dogs* then correctly predict *all cats*$\models$*several cats*. Interestingly, on the QN entailment task, neither our classifier trained on AN-N pairs nor the balAPinc method beat baseline methods. This suggests that our successful QN classifiers tap into vector properties beyond such relations as feature inclusion that those methods for nominal entailment rely upon.

Together, our experiments show that corpus-harvested DS representations of composite expressions such as ANs and QNs contain sufficient information to capture and generalize their inference patterns. This result brings DS closer to the central concerns of FS. In particular, the QN study is the first to our knowledge to show that DS vectors capture semantic properties not only of content words, but of an important class of function words (quantifying determiners) deeply studied in FS but of little interest until now in DS.

Besides these theoretical implications, our results are of practical import. First, our AN study presents a novel, practical method for detecting lexical entailment that reaches state-of-the-art performance with little or no manual intervention. Lexical entailment is in turn fundamental for constructing ontologies and other lexical resources (Buitelaar and Cimiano, 2008). Second, our QN study demonstrates that phrasal entailment can be automatically detected and thus paves the way to apply DS to advanced NLP tasks such as recognizing textual entailment (Dagan et al., 2009).

[1] In the sequel we will simply refer to a "quantifying determiner" as a "quantifier".

2 Background

2.1 Distributional semantics above the word level

DS models such as LSA (Landauer and Dumais, 1997) and HAL (Lund and Burgess, 1996) approximate the meaning of a word by a vector that summarizes its distribution in a corpus, for example by counting co-occurrences of the word with other words. Since semantically similar words tend to share similar contexts, DS has been very successful in tasks that require quantifying semantic similarity among words, such as synonym detection and concept clustering (Turney and Pantel, 2010).

Recently, there has been a flurry of interest in DS to model meaning composition: How can we derive the DS representation of a composite phrase from that of its constituents? Although the general focus in the area is to perform algebraic operations on word semantic vectors (Mitchell and Lapata, 2010), some researchers have also directly examined the corpus contexts of phrases. For example, Baldwin et al. (2003) studied vector extraction for phrases because they were interested in the decomposability of multiword expressions. Baroni and Zamparelli (2010) and Guevara (2010) look at corpus-harvested phrase vectors to learn composition functions that should derive such composite vectors automatically. Baroni and Zamparelli, in particular, showed qualitatively that directly corpus-harvested vectors for AN constructions are meaningful; for example, the vector of *young husband* has nearest neighbors *small son*, *small daughter* and *mistress*. Following up on this approach, we show here quantitatively that corpus-harvested AN vectors are also useful for detecting entailment. We find moreover distributional vectors informative and useful not only for phrases made of content words (such as ANs) but also for phrases containing functional elements, namely quantifying determiners.

2.2 Entailment from formal to distributional semantics

Entailment in FS To characterize the conditions under which a sentence is true, FS begins with the lexical meanings of the words in the sentence and builds up the meanings of larger and larger phrases until it arrives at the meaning of the whole sentence. The meanings throughout this

compositional process inhabit a variety of semantic domains, depending on the syntactic category of the expressions: typically, a sentence denotes a truth value (`true` or `false`) or truth conditions, a noun such as *cat* denotes a set of entities, and a quantifier phrase (QP) such as *all cats* denotes a set of sets of entities.

The entailment relation ($\models$) is a core notion of logic: it holds between one or more sentences and a sentence such that it cannot be that the former (antecedent) are true and the latter (consequent) is false. FS extends this notion from formal-logic sentences to natural-language expressions. By assigning meanings to parts of a sentence, FS allows defining entailment not only among sentences but also among words and phrases. Each semantic domain A has its own entailment relation $\models_A$. The entailment relation $\models_{\mathrm{S}}$ among sentences is the logical notion just described, whereas the entailment relations $\models_{\mathrm{N}}$ and $\models_{\mathrm{QP}}$ among nouns and quantifier phrases are the inclusion relations among sets of entities and sets of sets of entities respectively. Our results in Section 5 show that DS needs to treat $\models_{\mathrm{N}}$ and $\models_{\mathrm{QP}}$ differently as well.

Empirical, corpus-based perspectives on entailment Until recently, the corpus-based research tradition has studied entailment mostly at the word level, with applied goals such as classifying lexical relations and building taxonomic WordNet-like resources automatically. The most popular approach, first adopted by Hearst (1992), extracts lexical relations from patterns in large corpora. For instance, from the pattern N_1 *such as* N_2 one learns that $N_2 \models N_1$ (from *insects such as beetles*, derive *beetles* $\models$ *insects*). Several studies have refined and extended this approach (Pantel and Ravichandran, 2004; Snow et al., 2005; Snow et al., 2006; Turney, 2008).

While empirically very successful, the pattern-based method is mostly limited to single content words (or frequent content-word phrases). We are interested in entailment between phrases, where it is not obvious how to use lexico-syntactic patterns and cope with data sparsity. For instance, it seems hard to find a pattern that frequently connects one QP to another it entails, as in *all beetles PATTERN many beetles*. Hence, we aim to find a more general method and investigate whether DS vectors (whether corpus-harvested or compositionally derived) encode the information needed to account for phrasal entailment in a way that can be captured and generalized to unseen phrase pairs.

Rather recently, the study of sentential entailment has taken an empirical turn, thanks to the development of benchmarks for entailment systems. The FS definition of entailment has been modified by taking common sense into account. Instead of a relation from the truth of the consequent to the truth of the antecedent in any circumstance, the applied view looks at entailment in terms of plausibility: $\phi \models \psi$ if a human who reads (and trusts) ϕ would most likely infer that ψ is also true. Entailment systems have been compared under this new perspective in various evaluation campaigns, the best known being the Recognizing Textual Entailment (RTE) initiative (Dagan et al., 2009).

Most RTE systems are based on advanced NLP components, machine learning techniques, and/or syntactic transformations (Zanzotto et al., 2007; Kouleykov and Magnini, 2005). A few systems exploit deep FS analysis (Bos and Markert, 2006; Chambers et al., 2007). In particular, the FS results about QP properties that affect entailment have been exploited by Chambers et al, who complement a core broad-coverage system with a Natural Logic module to trade lower recall for higher precision. For instance, they exploit the monotonicity properties of *no* that cause the following reversal in entailment direction: *some beetles* $\models$ *some insects* but *no insects* $\models$ *no beetles*.

To investigate entailment step by step, we address here a much simpler and clearer type of entailment than the more complex notion taken up by the RTE community. While RTE is outside our present scope, we do focus on QP entailment as Natural Logic does. However, our evaluation differs from Chambers et al.'s, since we rely on general-purpose DS vectors as our only resource, and we look at phrase pairs with different quantifiers but the same noun. For instance, we aim to predict that *all beetles* $\models$ *many beetles* but *few beetles* $\not\models$ *all beetles*. QPs, of course, have many well-known semantic properties besides entailment; we leave their analysis to future study.

Entailment in DS Erk (2009) suggests that it may not be possible to induce lexical entailment directly from a vector space representation, but it is possible to encode the relation in this space after it has been derived through other means. On the other hand, recent studies (Geffet and Dagan,

2005; Kotlerman et al., 2010; Weeds et al., 2004) have pursued the intuition that entailment is the asymmetric ability of one term to "substitute" for another. For example, *baseball* contexts are also *sport* contexts but not *vice versa*, hence *baseball* is "narrower" than *sport* and *baseball* $\models$ *sport*. On this view, entailment between vectors corresponds to inclusion of contexts or features, and can be captured by asymmetric measures of distribution similarity. In particular, Kotlerman et al. (2010) carefully crafted the balAPinc measure (see Section 3.5 below). We adopt this measure because it has been shown to outperform others in several tasks that require lexical entailment information.

Like Kotlerman et al., we want to capture the entailment relation between vectors of features. However, we are interested in entailment not only between words but also between phrases, and we ask whether the DS view of entailment as feature inclusion, which captures entailment between nouns, also captures entailment between QPs. To this end, we complement balAPinc with a more flexible supervised classifier.

3 Data and methods

3.1 Semantic space

We construct distributional semantic vectors from the 2.83-billion-token concatenation of the British National Corpus (`http://www.natcorp.ox.ac.uk/`), WackyPedia and ukWaC (`http://wacky.sslmit.unibo.it/`). We tokenize and POS-tag this corpus, then lemmatize it with TreeTagger (Schmid, 1995) to merge singular and plural instances of words and phrases (*some dogs* is mapped to *some dog*).

We process the corpus in two steps to compute *semantic vectors* representing our *phrases of interest*. We use *phrases of interest* as a general term to refer to both multiword phrases and single words, and more precisely to: those AN and QN sequences that are in the data sets (see next subsections), the adjectives, quantifiers and nouns contained in those sequences, and the most frequent (9.8K) nouns and (8.1K) adjectives in the corpus. The first step is to count the content words (more precisely, the most frequent 9.8K nouns, 8.1K adjectives, and 9.6K verbs in the corpus) that occur in the same sentence as phrases of interest. In the second step, following standard practice, the co-occurrence counts are converted into *pointwise mutual information* (PMI) scores (Church and Hanks, 1990). The result of this step is a sparse matrix (with both positive and negative entries) with 48K rows (one per phrase of interest) and 27K columns (one per content word).

3.2 The AN $\models$ N data set

To characterize entailment between nouns using their semantic vectors, we need data exemplifying which noun entails which. This section introduces one cheap way to collect such a training data set exploiting semantic vectors for composed expressions, namely AN sequences. We rely on the linguistic fact that ANs share a syntactic category and semantic type with plain common nouns (*big cat* shares syntactic category and semantic type with *cat*). Furthermore, most adjectives are *restrictive* in the sense that, for every noun N, the AN sequence entails the N alone (every *big cat* is a *cat*). From a distributional point of view, the vector for an N should by construction include the information in the vector for an AN, given that the contexts where the AN occurs are a subset of the contexts where the N occurs (*cat* occurs in all the contexts where *big cat* occurs). This ideal inclusion suggests that the DS notion of lexical entailment as feature inclusion (see Section 2.2 above) should be reflected in the AN $\models$ N pattern.

Because most ANs entail their head Ns, we can create positive examples of AN $\models$ N without any manual inspection of the corpus: simply pair up the semantic vectors of ANs and Ns. Furthermore, because an AN usually does not entail another N, we can create negative examples ($AN_1 \not\models N_2$) just by randomly permuting the Ns. Of course, such unsupervised data would be slightly noisy, especially because some of the most frequent adjectives are not restrictive.

To collect cleaner data and to be sure that we are really examining the phenomenon of entailment, we took a mere few moments of manual effort to select the 256 restrictive adjectives from the most frequent 300 adjectives in the corpus. We then took the Cartesian product of these 256 adjectives with the 200 concrete nouns in the BLESS data set (Baroni and Lenci, 2011). Those nouns were chosen to avoid highly polysemous words. From the Cartesian product, we obtain a total of 1246 AN sequences, such as *big cat*, that occur more than 100 times in the corpus. These AN sequences encompass 190 of the 256 adjec-

tives and 128 of the 200 nouns.

The process results in 1246 positive instances of AN $\models$ N entailment, which we use as training data. To create a comparable amount of negative data, we randomly permuted the nouns in the positive instances to obtain pairs of $AN_1 \not\models N_2$ (e.g., *big cat* $\not\models$ *dog*). We manually double-checked that all positive and negative examples are correctly classified (2 of 1246 negative instances were removed, leaving 1244 negative training examples).

3.3 The lexical entailment $N_1 \models N_2$ data set

For testing data, we first listed all WordNet nouns in our corpus, then extracted hyponym-hypernym chains linking the first synsets of these nouns. For example, *pope* is found to entail *leader* because WordNet contains the chain *pope* $\rightarrow$ *spiritual_leader* $\rightarrow$ *leader*. Eliminating the 20 hypernyms with more than 180 hyponyms (mostly very abstract nouns such as *entity*, *object*, and *quality*) yields 9734 hyponym-hypernym pairs, encompassing 6402 nouns. Manually double-checking these pairs leaves us with 1385 positive instances of $N_1 \models N_2$ entailment.

We created the negative instances of again 1385 pairs by inverting 33% of the positive instances (from *pope* $\models$ *leader* to *leader* $\not\models$ *pope*), and by randomly shuffling the words across the positive instances. We also manually double-checked these pairs to make sure that they are not hyponym-hypernym pairs.

3.4 The $Q_1N \models Q_2N$ data set

We study 12 quantifiers: *all, both, each, either, every, few, many, most, much, no, several, some*. We took the Cartesian product of these quantifiers with the 6402 WordNet nouns described in Section 3.3. From this Cartesian product, we obtain a total of 28926 QN sequences, such as *every cat*, that occur at least 100 times in the corpus. These are our QN phrases of interest to which the procedure in Section 3.1 assigns a semantic vector.

Also, from the set of quantifier pairs (Q_1, Q_2) where $Q_1 \neq Q_2$, we identified 13 clear cases where $Q_1 \models Q_2$ and 17 clear cases where $Q_1 \not\models Q_2$. These 30 cases are listed in the first column of Table 1. For each of these 30 quantifier pairs (Q_1, Q_2), we enumerate those WordNet nouns N such that semantic vectors are available for both Q_1N and Q_2N (that is, both sequences occur in at least 100 times). Each such noun then gives rise to an instance of entailment ($Q_1N \models Q_2N$ if $Q_1 \models Q_2$; example: *many dogs* $\models$ *several dogs*) or non-entailment ($Q_1N \not\models Q_2N$ if $Q_1 \not\models Q_2$; example: *many dogs* $\not\models$ *most dogs*). The number of QN pairs that each quantifier pair gives rise to in this way is listed in the second column of Table 1. As shown there, we have a total of 7537 positive instances and 8455 negative instances of QN entailment.

Quantifier pair	Instances	Correct
all $\models$ some	1054	1044 (99%)
all $\models$ several	557	550 (99%)
each $\models$ some	656	647 (99%)
all $\models$ many	873	772 (88%)
much $\models$ some	248	217 (88%)
every $\models$ many	460	400 (87%)
many $\models$ some	951	822 (86%)
all $\models$ most	465	393 (85%)
several $\models$ some	580	439 (76%)
both $\models$ some	573	322 (56%)
many $\models$ several	594	113 (19%)
most $\models$ many	463	84 (18%)
both $\models$ either	63	1 (2%)
Subtotal	*7537*	*5804 (77%)*
some $\not\models$ every	484	481 (99%)
several $\not\models$ all	557	553 (99%)
several $\not\models$ every	378	375 (99%)
some $\not\models$ all	1054	1043 (99%)
many $\not\models$ every	460	452 (98%)
some $\not\models$ each	656	640 (98%)
few $\not\models$ all	157	153 (97%)
many $\not\models$ all	873	843 (97%)
both $\not\models$ most	369	347 (94%)
several $\not\models$ few	143	134 (94%)
both $\not\models$ many	541	397 (73%)
many $\not\models$ most	463	300 (65%)
either $\not\models$ both	63	39 (62%)
many $\not\models$ no	714	369 (52%)
some $\not\models$ many	951	468 (49%)
few $\not\models$ many	161	33 (20%)
both $\not\models$ several	431	63 (15%)
Subtotal	*8455*	*6690 (79%)*
Total	*15992*	*12494 (78%)*

Table 1: Entailing and non-entailing quantifier pairs with number of instances per pair (Section 3.4) and $SVM_{pair\text{-}out}$ performance breakdown (Section 5).

3.5 Classification methods

We consider two methods to classify candidate pairs as entailing or non-entailing, the balAPinc measure of Kotlerman et al. (2010) and a standard Support Vector Machine (SVM) classifier.

balAPinc As discussed in Section 2.2, balAPinc is optimized to capture a relation of feature inclusion between the narrower (entailing) and broader (entailed) terms, while capturing other intuitions about the relative relevance of features. balAPinc averages two terms, APinc and LIN. APinc is given by:

$$\mathrm{APinc}(u \models v) = \frac{\sum_{r=1}^{|F_u|}\big(P(r) \cdot rel'(f_r)\big)}{|F_u|}$$

APinc is a version of the Average Precision measure from Information Retrieval tailored to lexical inclusion. Given vectors F_u and F_v representing the dimensions with positive PMI values in the semantic vectors of the candidate pair $u \models v$, the idea is that we want the features (that is, vector dimensions) that have larger values in F_u to also have large values in F_v (the opposite does not matter because it is u that should be included in v, not *vice versa*). The F_u features are ranked according to their PMI value so that f_r is the feature in F_u with rank r, i.e., r-th highest PMI. Then the sum of the product of the two terms $P(r)$ and $rel'(f_r)$ across the features in F_u is computed. The first term is the precision at r, which is higher when highly ranked u features are present in F_v as well. The relevance term $rel'(f_r)$ is higher when the feature f_r in F_u also appears in F_v with a high rank. (See Kotlerman et al. for how $P(r)$ and $rel'(f_r)$ are computed.) The resulting score is normalized by dividing by the entailing vector size $|F_u|$ (in accordance with the idea that having more v features should not hurt because the u features should be included in the v features, not *vice versa*).

To balance the potentially excessive asymmetry of APinc towards the features of the antecedent, Kotlerman et al. average it with LIN, the widely used symmetric measure of distributional similarity proposed by Lin (1998):

$$\mathrm{LIN}(u, v) = \frac{\sum_{f \in F_u \cap F_v}[w_u(f) + w_v(f)]}{\sum_{f \in F_u} w_u(f) + \sum_{f \in F_v} w_v(f)}$$

LIN essentially measures feature vector overlap. The positive PMI values $w_u(f)$ and $w_v(f)$ of a feature f in F_u and F_v are summed across those features that are positive in both vectors, normalizing by the cumulative positive PMI mass in both vectors. Finally, balAPinc is the geometric average of APinc and LIN:

$$\mathrm{balAPinc}(u \models v) = \sqrt{\mathrm{APinc}(u \models v) \cdot \mathrm{LIN}(u, v)}$$

To adapt balAPinc to recognize entailment, we must select a threshold t above which we classify a pair as entailing. In the experiments below, we explore two approaches. In balAPinc$_{\mathrm{upper}}$, we optimize the threshold directly on the test data, by setting t to maximize the F-measure on the test set. This gives us an upper bound on how well balAPinc could perform on the test set (but note that optimizing F does not necessarily translate into a good accuracy performance, as clearly illustrated by Table 3 below). In balAPinc$_{\mathrm{AN} \models \mathrm{N}}$, we use the AN $\models$ N data set as training data and pick the t that maximizes F on this training set.

We use the balAPinc measure as a reference point because, on the evidence provided by Kotlerman et al., it is the state of the art in various tasks related to lexical entailment. We recognize however that it is somewhat complex and specifically tuned to capturing the relation of feature inclusion. Consequently, we also experiment with a more flexible classifier, which can detect other systematic properties of vectors in an entailment relation. We present this classifier next.

SVM Support vector machines are widely used high-performance discriminative classifiers that find the hyperplane providing the best separation between negative and positive instances (Cristianini and Shawe-Taylor, 2000). Our SVM classifiers are trained and tested using Weka 3 and LIBSVM 2.8 (Chang and Lin, 2011). We use the default polynomial kernel ($(u \cdot v/600)^3$) with ϵ (tolerance of termination criterion) set to 1.6. This value was tuned on the AN $\models$ N data set, which we never use for testing. In the same initial tuning experiments on the AN $\models$ N data set, SVM outperformed decision trees, naive Bayes, and k-nearest neighbors.

We feed each potential entailment pair to SVM by concatenating the two vectors representing the antecedent and consequent expressions.[2] However, for efficiency and to mitigate data sparseness, we reduce the dimensionality of the semantic vectors to 300 columns using Singular Value Decomposition (SVD) before feeding them to the classifier.[3] Because the SVD-reduced semantic

[2] We have tried also to represent a pair by subtracting and by dividing the two vectors. The concatenation operation gave more successful results.

[3] To keep a manageable parameter space, we picked 300 columns without tuning. This is the best value reported in many earlier studies, including classic LSA. Since SVD sometimes improves the semantic space (Landauer and Du-

vectors occupy a 300-dimensional space, the entailment pairs occupy a 600-dimensional space.

An SVM with a polynomial kernel takes into account not only individual input features but also their interactions (Manning et al., 2008, chapter 15). Thus, our classifier can capture not just properties of individual dimensions of the antecedent and consequent pairs, but also properties of their combinations (e.g., the product of the first dimensions of the antecedent and the consequent). We conjecture that this property of SVMs is fundamental to their success at detecting entailment, where relations between the antecedent and the consequent should matter more than their independent characteristics.

4 Predicting lexical entailment from AN $\models$ N evidence

Since the contexts of AN must be a subset of the contexts of N, semantic vectors harvested from AN phrases and their head Ns are by construction in an inclusion relation. The first experiment shows that these vectors constitute excellent training data to discover entailment between nouns. This suggests that the vector pairs representing entailment between nouns are also in an inclusion relation, supporting the conjectures of Kotlerman et al. (2010) and others.

Table 2 reports the results we obtained with $\text{balAPinc}_{\text{upper}}$, $\text{balAPinc}_{\text{AN} \models \text{N}}$ (Section 3.5) and $\text{SVM}_{\text{AN} \models \text{N}}$ (the SVM classifier trained on the AN $\models$ N data). As an upper bound for methods that generalize from AN $\models$ N, we also report the performance of SVM trained with 10-fold cross-validation on the $\text{N}_1 \models \text{N}_2$ data themselves ($\text{SVM}_{\text{upper}}$). Finally, we tried two baseline classifiers. The first baseline ($\text{fq}(\text{N}_1) < \text{fq}(\text{N}_2)$) guesses entailment if the first word is less frequent than the second. The second ($\cos(\text{N}_1, \text{N}_2)$) applies a threshold (determined on the test set) to the cosine similarity of the pair. The results of these baselines shown in Table 2 use SVD; those without SVD are similar. Both baselines outperformed more trivial methods such as random guessing or fixed response, but they performed significantly worse than SVM and balAPinc.

Both methods that generalize entailment from AN $\models$ N to $\text{N}_1 \models \text{N}_2$ perform well, with 70% accuracy on the test set, which is balanced between positive and negative instances. Interestingly, the balAPinc decision thresholds tuned on the AN $\models$ N set and on the test data are very close (0.26 vs. 0.24), resulting in very similar performance for $\text{balAPinc}_{\text{AN} \models \text{N}}$ and $\text{balAPinc}_{\text{upper}}$. This suggests that the relation captured by balAPinc on the phrasal entailment training data is indeed the same that the measure captures when applied to lexical entailment data.

	P	R	F	Accuracy (95% C.I.)
$\text{SVM}_{\text{upper}}$	88.6	88.6	88.5	88.6 (87.3–89.7)
$\text{balAPinc}_{\text{AN} \models \text{N}}$	65.2	87.5	74.7	70.4 (68.7–72.1)
$\text{balAPinc}_{\text{upper}}$	64.4	90.0	75.1	70.1 (68.4–71.8)
$\text{SVM}_{\text{AN} \models \text{N}}$	69.3	69.3	69.3	69.3 (67.6–71.0)
$\cos(\text{N}_1, \text{N}_2)$	57.7	57.6	57.5	57.6 (55.8–59.5)
$\text{fq}(\text{N}_1) < \text{fq}(\text{N}_2)$	52.1	52.1	51.8	53.3 (51.4–55.2)

Table 2: Detecting lexical entailment. Results ranked by accuracy and expressed as percentages. 95% confidence intervals around accuracy calculated by binomial exact tests.

The success of this first experiment shows that the entailment relation present in the distributional representation of AN phrases and their head Ns transfers to lexical entailment (entailment among Ns). Most importantly, this result demonstrates that the semantic vectors of composite expressions (such as ANs) are useful for lexical entailment. Moreover, the result is in accordance with the view of FS, that ANs and Ns have the same semantic type, and thus they enter entailment relations of the same kind. Finally, the hypothesis that entailment among nouns is reflected by distributional inclusion among their semantic vectors (Kotlerman et al., 2010) is supported both by the successful generalization of the SVM classifier trained on AN $\models$ N pairs and by the good performance of the balAPinc measure.

5 Generalizing QN entailment

The second study is somewhat more ambitious, as it aims to capture and generalize the entailment relation between QPs (of shape QN) using only the corpus-harvested semantic vectors representing these phrases as evidence. We are thus first and foremost interested in testing whether these vectors encode information that can help a power-

mais, 1997; Rapp, 2003; Schütze, 1997), we tried balAPinc on the SVD-reduced vectors as well, but results were consistently worse than with PMI vectors.

	P	R	F	Accuracy (95% C.I.)
$\text{SVM}_{\text{pair-out}}$	76.7	77.0	76.8	78.1 (77.5–78.8)
$\text{SVM}_{\text{quantifier-out}}$	70.1	65.3	68.0	71.0 (70.3–71.7)
$\text{SVM}^{\text{Q}}_{\text{pair-out}}$	67.9	69.8	68.9	70.2 (69.5–70.9)
$\text{SVM}^{\text{Q}}_{\text{quantifier-out}}$	53.3	52.9	53.1	56.0 (55.2–56.8)
$\cos(\text{QN}_1, \text{QN}_2)$	52.9	52.3	52.3	53.1 (52.3–53.9)
$\text{balAPinc}_{\text{AN} \models \text{N}}$	46.7	5.6	10.0	52.5 (51.7–53.3)
$\text{SVM}_{\text{AN} \models \text{N}}$	2.8	42.9	5.2	52.4 (51.7–53.2)
$\text{fq}(\text{QN}_1) < \text{fq}(\text{QN}_2)$	51.0	47.4	49.1	50.2 (49.4–51.0)
$\text{balAPinc}_{\text{upper}}$	47.1	100	64.1	47.2 (46.4–47.9)

Table 3: Detecting quantifier entailment. Results ranked by accuracy and expressed as percentages. 95% confidence intervals around accuracy calculated by binomial exact tests.

ful classifier, such as SVM, to detect entailment.

To abstract away from lexical or other effects linked to a specific quantifier, we consider two challenging training and testing regimes. In the first ($\text{SVM}_{\text{pair-out}}$), we hold out one quantifier pair as testing data and use the other 29 pairs in Table 1 as training data. Thus, for example, the classifier must discover *all dogs* $\models$ *some dogs* without seeing any *all N* $\models$ *some N* instance in the training data. In the second ($\text{SVM}_{\text{quantifier-out}}$), we hold out one of the 12 quantifiers as testing data (that is, hold out every pair involving a certain quantifier) and use the rest as training data. For example, the quantifier must guess *all dogs* $\models$ *some dogs* without ever seeing *all* in the training data. We expect the second training regime to be more difficult, not just because there is less training data, but also because the trained classifier is tested on a quantifier that it has never encountered within any training QN sequence.[4]

Table 3 reports the results for $\text{SVM}_{\text{pair-out}}$ and $\text{SVM}_{\text{quantifier-out}}$, as well as for the methods we tried in the lexical entailment experiments. (As in the first study, the frequency- and cosine-based baselines are only slightly better overall than more trivial baselines.) We consider moreover an alternative approach that ignores the noun altogether and uses vectors for the quantifiers only (e.g., the decision about *all dogs* $\models$ *some dogs* considers the corpus-derived *all* and *some* vectors only). The models resulting from this Q-only strategy are marked with the superscript Q in the table.

The results confirm clearly that semantic vectors for QNs contain enough information to allow a classifier to detect entailment: $\text{SVM}_{\text{quantifier-out}}$ performs as well as the lexical entailment classifiers of our first study, and $\text{SVM}_{\text{pair-out}}$ does even better. This success is especially impressive given our challenging training and testing regimes.

In contrast to the first study, now $\text{SVM}_{\text{AN} \models \text{N}}$, the classifier trained on the AN $\models$ N data set, and balAPinc perform no better than the baselines. (Here $\text{balAPinc}_{\text{upper}}$ and $\text{balAPinc}_{\text{AN} \models \text{N}}$ pick very different thresholds: the first settling on a very low $t = 0.01$, whereas for the second $t = 0.26$.) As predicted by FS (see Section 2.2 above), noun-level entailment does not generalize to quantifier phrase entailment, since the two structures have different semantic types, corresponding to different kinds of entailment relations. Moreover, the failure of balAPinc suggests that, whatever evidence the SVMs rely upon, it is not simple feature inclusion.

Interestingly, even the Q vectors alone encode enough information to capture entailment above chance. Still, the huge drop in performance from $\text{SVM}^{\text{Q}}_{\text{pair-out}}$ to $\text{SVM}^{\text{Q}}_{\text{quantifier-out}}$ suggests that the Q-only method learned ad-hoc properties that do not generalize (e.g., "*all* entails every Q_2").

Tables 1 and 4 break down the SVM results by (pairs of) quantifiers. We highlight the remarkable dichotomy in Table 4 between the good performance on the universal-like quantifiers (*each*, *every*, *all*, *much*) and the poor performance on the existential-like ones (*some*, *no*, *both*, *either*).

In sum, the QN experiments show that semantic vectors contain enough information to detect a logical relation such as entailment not only between words, but also between phrases containing quantifiers that determine their entailment relation. While a flexible classifier such as SVM performs this task well, neither measuring feature inclusion nor generalizing nominal entailment works. SVMs are evidently tapping into other properties of the vectors.

[4] In our initial experiments, we added negative entailment instances by blindly permuting the nouns, under the assumption that Q_1N_1 typically does not entail Q_2N_2 when $Q_1 \neq Q_2$ and $N_1 \neq N_2$. These additional instances turned out to be much easier to classify: adding an equal proportion of them to the training data and testing data, such that the number of instances where $N_1 = N_2$ and where $N_1 \neq N_2$ is equal, reduced every error rate roughly by half. The reported results do not involve these additional instances.

Quantifier	Instances		Correct		
	$\models$	$\not\models$	$\models$	$\not\models$	
each	656	656	649	637	(98%)
every	460	1322	402	1293	(95%)
much	248	0	216	0	(87%)
all	2949	2641	2011	2494	(81%)
several	1731	1509	1302	1267	(79%)
many	3341	4163	2349	3443	(77%)
few	0	461	0	311	(67%)
most	928	832	549	511	(60%)
some	4062	3145	1780	2190	(55%)
no	0	714	0	380	(53%)
both	636	1404	589	303	(44%)
either	63	63	2	41	(34%)
Total	*15074*	*16910*	*9849*	*12870*	*(71%)*

Table 4: Breakdown of results with leaving-one-quantifier-out ($SVM_{\text{quantifier-out}}$) training regime.

6 Conclusion

Our main results are as follows.

1. Corpus-harvested semantic vectors representing adjective-noun constructions and their heads encode a relation of entailment that can be exploited to train a classifier to detect lexical entailment. In particular, a relation of feature inclusion between the narrower antecedent and broader consequent terms captures both AN $\models$ N and $N_1 \models N_2$ entailment.

2. The semantic vectors of quantifier-noun constructions also encode information sufficient to learn an entailment relation that generalizes to QNs containing quantifiers that were not seen during training.

3. Neither the entailment information encoded in AN $\models$ N vectors nor the balAPinc measure generalizes well to entailment detection in QNs. This result suggests that QN vectors encode a different kind of entailment, as also suggested by type distinctions in Formal Semantics.

In future work, we want first of all to conduct an analysis of the features in the $Q_1N \models Q_2N$ vectors that are crucially exploited by our successful entailment recognizers, in order to understand which characteristics of entailment are encoded in these vectors.

Very importantly, instead of extracting vectors representing phrases directly from the corpus, we intend to derive them by compositional operations proposed in the literature (see Section 2.1 above). We will look for composition methods producing vector representations of composite expressions that are as good as (or better than) vectors directly extracted from the corpus at encoding entailment.

Finally, we would like to evaluate our entailment detection strategies for larger phrases and sentences, possibly containing multiple quantifiers, and eventually embed them as core components of an RTE system.

Acknowledgments

We thank the Erasmus Mundus EMLCT Program for the student and visiting scholar grants to the third and fourth author, respectively. The first two authors are partially funded by the ERC 2011 Starting Independent Research Grant supporting the COMPOSES project (nr. 283554). We are grateful to Gemma Boleda, Louise McNally, and the anonymous reviewers for valuable comments, and to Ido Dagan for important insights into entailment from an empirical point of view.

References

Timothy Baldwin, Colin Bannard, Takaaki Tanaka, and Dominic Widdows. 2003. An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions*, pages 89–96.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of EMNLP*, pages 1183–1193, Boston, MA.

Johan Bos and Katja Markert. 2006. When logical inference helps determining textual entailment (and when it doesn't. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Paul Buitelaar and Philipp Cimiano. 2008. *Bridging the Gap between Text and Knowledge*. IOS, Amsterdam.

Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine de Marneffe, Daniel Ramage, Eric Yeh,

and Christopher D. Manning. 2007. Learning alignments and leveraging natural logic. In *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27.

Kenneth Church and Peter Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Nello Cristianini and John Shawe-Taylor. 2000. *An introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, Cambridge.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2009. Recognizing textual entailment: rational, evaluation and approaches. *Natural Language Engineering*, 15:459–476.

Katrin Erk. 2009. Supporting inferences in semantic space: representing words as regions. In *Proceedings of IWCS*, pages 104–115, Tilburg, Netherlands.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of ACL*, pages 107–114, Ann Arbor, MI.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of EMNLP*, pages 1395–1404, Edinburgh.

Emiliano Guevara. 2010. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of the ACL GEMS Workshop*, pages 33–37, Uppsala, Sweden.

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING*, pages 539–545, Nantes, France.

Irene Heim and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell, Oxford.

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

Milen Kouleykov and Bernardo Magnini. 2005. Tree edit sistance for textual entailment. In *Proceedings of RALNP-2005, International Conference on Recent Advances in Natural Language Processing*, pages 271–278.

Thomas Landauer and Susan 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.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of ICML*, pages 296–304, Madison, WI, USA.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods*, 28:203–208.

Chris Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

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

Patrick Pantel and Deepak Ravichandran. 2004. Automatically labeliing semantic classes. In *Proceedings of HLT-NAACL 2004*, pages 321–328.

Reinhard Rapp. 2003. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the 9th MT Summit*, pages 315–322, New Orleans, LA.

Magnus Sahlgren. 2006. *The Word-Space Model*. Dissertation, Stockholm University.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the EACL-SIGDAT Workshop*, Dublin, Ireland.

Hinrich Schütze. 1997. *Ambiguity Resolution in Natural Language Learning*. CSLI, Stanford, CA.

Rion Snow, Daniel Juravsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Proceedings of NIPS 17*.

Rion Snow, Daniel Juravsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of ACL 2006*, pages 801–808.

Richmond H. Thomason, editor. 1974. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New York.

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

Peter Turney. 2008. A uniform approach to analogies, synonyms, antonyms and associations. In *Proceedings of COLING*, pages 905–912, Manchester, UK.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference of Computational Linguistics, COLING-2004*, pages 1015–1021.

Fabio M. Zanzotto, Marco Pennacchiotti, and Alessandro Moschitti. 2007. Shallow semantics in fast textual entailment rule learners. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Maayan Zhitomirsky-Geffet and Ido Dagan. 2010. Bootstrapping distributional feature vector quality. *Computational Linguistics*, 35(3):435–461.