So similar and yet incompatible: Toward automated identification of semantically compatible words

Germán Kruszewski and **Marco Baroni**
Center for Mind/Brain Sciences (University of Trento, Italy)
(german.kruszewski|marco.baroni)@unitn.it

Abstract

We introduce the challenge of detecting *semantically compatible* words, that is, words that can potentially refer to the same thing (*cat* and *hindrance* are compatible, *cat* and *dog* are not), arguing for its central role in many semantic tasks. We present a publicly available data-set of human compatibility ratings, and a neural-network model that takes distributional embeddings of words as input and learns alternative embeddings that perform the compatibility detection task quite well.

1 Introduction

Vectors encoding distributional information extracted from large text corpora provide very effective estimates of semantic similarity or, more generally, relatedness between words (Clark, 2015; Erk, 2012; Turney and Pantel, 2010). Semantic relatedness is undoubtedly a core property of word understanding, and indeed current vector-based *distributional semantic models* (DSMs) provide an impressive approximation to human judgments in many tasks (Baroni et al., 2014). However, relatedness alone is too general a notion to truly capture the nuances of human conceptual knowledge. The terms *animal*, *puppy*, and *cat* are all closely related to *dog*, but the nature of their relation is very different, each affording different inferences: If you tell me that Fido is a dog, I will also conclude that he's an animal, that he is not a cat, and that he might or might not be a puppy.

The previous examples hint at a fundamental semantic property that is only partially linked to relatedness, namely *compatibility*, that we define, for our current purposes, as follows: *Linguistic expressions w_1 and w_2 are compatible iff, in a reasonably normal state of affairs, they can both truthfully refer to the same thing. If they cannot, then they are incompatible*. We realize that the notion of a "reasonably normal sate of affairs" is dangerously vague, but we want to exclude science-fiction scenarios in which dogs mutate into cats. And we use *thing* as a catch-all term for anything words (or other linguistic expressions) can refer to (entities, events, collections, etc.).

The notions of compatibility and incompatibility have been introduced in theoretical semantics before (Cruse, 1986; Murphy, 2010). The definition that we give here for compatibility is related, but different from the one by Cruse. For example, subsuming pairs are out of the scope of compatibility under his definition, whereas we include them. Murphy defines incompatibility similarly to us, but she does not define compatibility. We are not aware, on the other hand, of any earlier systematic attempt to study the phenomenon empirically, nor to model it computationally.

In general, compatible terms will be semantically related (*dog* and *animal*). However, relatedness does not suffice: many semantically related, even very similar terms are not compatible (*dog* and *cat*). Relatedness is not even a necessary condition: A *husband* can be a *hindrance* in an all-too-normal state of affairs, but the concepts of husband and hindrance are not semantically close. Moreover, compatibility does not reduce to (a set of) more commonly studied semantic relations. While it relates to hy-

pernymy, synonymy and co-hyponymy, there are cases, such as *husband*/*hindrance*, that do not naturally map to any of these relations. Also, although many incompatibles among closely related pairs are co-hyponyms, this is not necessarily the case: You cannot be both a *dog* and a *cat*, but you can be a *violinist* and a *drummer*.

We argue that, since knowing what's compatible plays a central role in human semantic reasoning, algorithms that determine compatibility automatically will help in many domains that require human-like semantic knowledge. Most obviously, compatibility is a necessary (although not sufficient) prerequisite for coreference. *Dog* and *puppy* could belong to the same coreference chain, whereas *dog* and *cat* do not. We conjecture that the relatively disappointing performance of DSMs in support of coreference resolution (Poesio et al., 2010) is at least partially due to the inability of standard DSMs to distinguish compatible and incompatible terms. Compatibility is also central to recognizing entailment (and contradiction): Standard DSMs are of relatively little use in recognizing entailment as they treat antonymous, contradictory words such as *dead* and *alive* as highly related (Adel and Schütze, 2014; Mohammad et al., 2013), with catastrophic results for the inferences that can be drawn (antonyms are just the tip of the incompatibility iceberg: *dog* and *cat* are not antonyms, but one still contradicts the other). Knowing what's compatible might also help in tasks that require recognizing (distant) paraphrases, such as question answering, document summarization or even machine translation (*the violinist also played the drum* might corefer with *the drummer also played the violin*, whereas *the dog was killed* and *the cat was killed* must refer to different events). Other applications could include modeling semantic plausibility of a nominal phrase (Vecchi et al., 2011; Lynott and Connell, 2009), where the goal is to accept expressions like *coastal mosquito*, but reject *parlamentary tomato*. Finally, the notion of incompatibility relates to (certain kinds of) negation. Negation is notoriously difficult to model with DSMs (Hermann et al., 2013), and compatibility might offer a new angle into it.

In this paper, we introduce a new, large benchmark to evaluate computational models on compatibility detection. We then present a supervised neural-network based model that takes distributional semantic vectors as input and embeds them into a space that is optimized for compatibility detection. The model performs significantly better than direct DSM relatedness, and achieves high scores in absolute terms.

2 The compatibility benchmark

We started the benchmark construction by manually assembling a list of 299 words including mostly concrete, basic-level concepts picked from categories where taxonomically close terms tend to be incompatible (e.g., biological classes such as animals and vegetables), as well as from categories that are more compatibility-prone (kinship terms, professions), or somewhere in the middle (tools, places). The list also included category names at different levels of abstraction (*creature*, *animal*, *carnivore*...), as well as some terms that were expected to be of high general compatibility (*hindrance*, *expert*, *companion*...). By randomly coupling words from this list, we generated pairs that should reflect a wide range of compatibility patterns (compatible and incompatible coordinate terms, words in an entailment relation, dissimilar but compatible, dissimilar and incompatible, etc.).[1] We generated about 18K such random pairs.

We used a subset of about 3K pairs in a pilot study on the CrowdFlower[2] crowd-sourcing platforms, in which we asked participants to annotate them for compatibility either as a yes/no judgment accompanied by a confidence rating, or on a 7-point scale. Correlation between mean binary and ordinal ratings was extremely high ($>$0.95), so we decided to adopt the potentially more precise, albeit more noisy, 7-point scale. Confidence judgments (median: 6.6/7), participant agreement and sanity checks on obvious cases confirmed that the raters understood the task well and produced the expected judgments consistently.

We thus launched a larger CrowdFlower survey,

[1] We realize that the resulting pairs might not resemble the natural distribution of compatibility decisions that an average person might encounter in daily life. However, the fact that (as we show below) subjects were highly consistent in judging the items proves that the data reflect genuine shared semantic knowledge a computational model should be able to capture.

[2] http://www.crowdflower.com

asking participants to rate pairs on a 7-point scale by answering the following question: "How much do you agree with the statement that <*word1*> and <*word2*> can refer to the same thing, animal or person?" We asked the judges to consider real-life scenarios and fairly ordinary circumstances; in case of ambiguity, they were asked to choose the sense that would make the pair compatible, as long as it was sufficiently common. 20 control items with obvious choices (e.g. *drummer/ant* - *writer/father*) were inserted to exclude raters that did not perform the task seriously. We paid close attention to contributors' feedback, correcting dubious controls. For example, we removed *bucket/chair*, since one contributor pointed out that you could turn a bucket upside down and use it as a chair.[3] In this way, we obtained usable annotation for 17973 pairs, each rated by 10 participants[4]. The average standard deviation was as as low as 0.70, compared to the standard deviation of a uniformly distributed multinomial distribution, which amounts to 1.8. As expected, ratings were highly skewed as most random pairs are incompatible: the median is 1.10 (with a standard deviation of 1.81). Yet, the overall distribution is bimodal, peaking at the two ends of the scale.

In order to be able to phrase (in)compatibility detection not only in continuous terms, but also as dichotomous tasks, we further produced a list of unambiguously (in)compatible pairs from the ends of the rating scale. Specifically, we manually inspected a subset of the list (before any computational simulation was run), and picked a mean 3.7 rating (exclusive) as minimum value for compatible pairs, and 1.6 (inclusive) as maximum score for incompatible ones. The number of problematic cases above/below these thresholds was absolutely negligible. We thus coded the data set by classifying the 2,933 pairs above the first threshold as compatible (e.g., *expert/criminal*, *hill/obstacle*, *snake/vermin*), the 12,669 pairs below the second as incompatible (e.g., *bottle/plate*, *cheetah/queen*), and the remainder as neither.

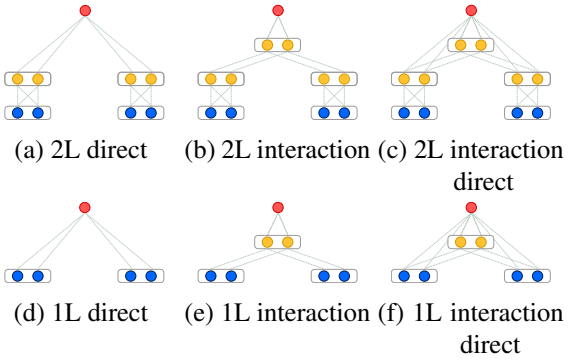


Figure 1: Schematic representation of the models

3 Models

We take DSM vectors as input, since they provide us with semantically rich word representations, and seek to induce a compatibility measure by learning the parameters of a model in a supervised manner. In particular, we used the word vectors publicly available at `http://clic.cimec.unitn.it/composes/semantic-vectors.html`. These vectors, extracted with the word2vec toolkit (Mikolov et al., 2013) from a 3B token corpus, were shown by Baroni et al. (2014) to produce near-state-of-the-art performance on a variety of semantic tasks.

We hypothesized that the interaction between a simple set of features (induced from the distributional ones) should account for a large portion of compatibility patterns. For example, human roles would typically be compatible (*classmate/friend*), whereas two animals would probably be incompatible (*iguana/zebra*). The model should thus be able to learn features associated to such classes, and compatibility rules associated to their interaction (e.g., if both w_1 and w_2 have large values for a *human* feature, compatibility is more likely). We incorporated this insight into the **2L interaction** neural network illustrated in Figure 1b. This network takes the distributional representations of the words in a pair, transforms them into new feature vectors by means of a mapping that is shared by both inputs, constructs the vector of pairwise interactions between the induced features, and finally uses the weighted combination of the latter to produce a real-number

[3] We also were surprised to learn that drummer ants actually exist. Yet, in that case we decided to keep the control item since, under the most common sense of *drummer*, and in ordinary circumstances, ants cannot be drummers.

[4] The guidlienes provided to the participants and the collected data set are available at: `http://clic.cimec.unitn.it/composes/`

score.

We considered then some variations of the 2L interaction model, to investigate the importance of each of its components. In **2L direct** (Figure 1a), we removed the interaction layer, making the model score a weighted combination of the mapped vectors. The **2L interaction direct** model (Figure 1c) computes the final score through a weighted combination of both the mapped representations and their interaction vector. The **1L** models (Figures 1d, 1e and 1f) are analogous to the corresponding 2L models, but removing the feature mapping layer, thus operating directly on the distributional vectors.

4 Experiments

Since compatibility is a symmetric relation, we first duplicated each pair in the benchmark by swapping the two words. We then split it into training, testing and development sections. To make the task more challenging, we enforced disjoint vocabularies in each of them. For example, *drummer* only occurs in the training set, while *ant*, only in the test set. We use about 1/10th of the vocabulary (29 words) on the development set and the rest was split equally between train and test (135 words each). The resulting partitions contain 7,228 (train), 7,336 (test) and 312 (development) pairs, respectively.

To train the models, we used the scores they generate in three sub-tasks: approximation of average ratings, classification of compatibles and classification of incompatibles. We used mean square error as cost function for the first sub-task, cross-entropy for the latter two.

We implemented the models in Torch7 (Collobert et al., 2011).[5] We trained them for 120 epochs with adagrad, with a batch size of 150 items and adopting an emphasizing scheme (LeCun et al., 2012), where compatibles, incompatibles and middle-ground items appear in equal proportions. We fixed hidden-layer size to 100 dimensions, while we tuned a coefficient for a L2-norm regularization term on the development data.

We evaluated the models ability to predict human compatibility ratings as well as to detect compatible and incompatible items.

[5] We make the code available at `https://github.com/germank/compatibility-naacl2015`

	corr.	comp.			incomp.		
Model	r	P	R	F1	P	R	F1
1L direct	50	59	55	57	80	83	72
1L interaction	51	50	61	55	80	77	79
1L int. direct	49	52	57	54	80	79	80
2L direct	49	51	58	54	81	79	80
2L interaction	**72**	76	58	**66**	84	90	**87**
2L int. direct	67	71	58	64	82	85	84
1L mono	35	31	57	41	79	77	78
2L mono	35	32	64	43	80	72	76
Cosine	36	29	58	38	78	71	74

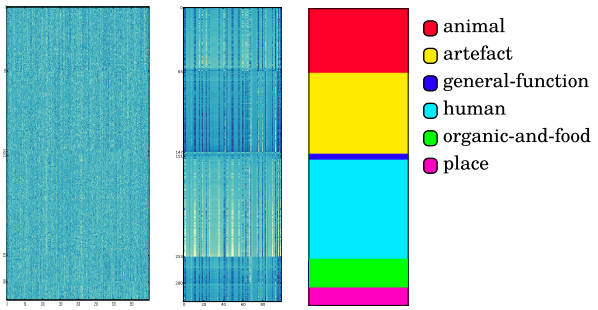
Table 1: Experimental results. Correlation with human ratings measured by Pearson r. (In)compatibility detection scored by the F1 measure.

We compared the supervised measures to the cosine of pairs directly represented by their DSM vectors (with thresholds tuned on the training set). We expected this baseline to fare relatively well on incompatibility detection, since many of our randomly generated pairs were both incompatible and dissimilar (e.g., *bag/bus*).

Also, we controlled for the portion of the data that can be accounted just by looking at one of the words of the relation (for example, the presence of a word might indicate that the relation is incompatible). To this end, we included two models that look at only one of the words in the pair. **1L mono** is a logistic regression model that only looks at the first word of the pair while **2L mono** is an analogous neural network with one hidden layer.

Results are reported in Table 1. As it can be seen, all the supervised models from Figure 1 strongly outperform the cosine (that, as expected, is nevertheless quite good at detecting incompatibles). Also, they outperform the mono models (with the only exception of 1L direct on incompatibility), showing that the data they account for cannot be reduced to properties of individual lexical items. Importantly, the 2L interaction model is way ahead of all other models, confirming our expectations.

To gain some insight into the features learned by the best model, we labeled the words of our input vocabulary with one of the following general category tags: *animal*, *artefact*, *general-function*, *human*, *organic-and-food* and *place*. The distribution



(a) Input vectors (b) Mapped vectors (c) Categories

Figure 2: Heatmap visualization of original DSM features and features learned by the mapping function of the 2L interaction model.

of the vocabulary across the labels is shown in Figure 2c. If we plot the input distributional vectors so that words tagged with the same category are adjacent to each other, and categories arranged as in Figure 2c, we obtain the heatmap in Figure 2a, where no obvious pattern emerges. If instead we plot the output vectors of 2L interaction mapping in the same way, we obtain the heatmap in Figure 2b. It is evident that the mapping produces vectors that are similar within most categories, and very different across them. Thus, the 2L interaction model clearly learned the relevance of general categories in capturing compatibility judgments. The fact that this model produced the best results hints at the importance of exploiting this source of information, confirming the intuition we used in designing it, that compatibility can be characterized by a combination of general relatedness and category-specific cues.

Finally, we explored to what extent the data can be accounted by co-hyponymy, an idea briefly introduced in the introductory discussion of Section 1. For simplicity purposes, we take the same category tags we just introduced as a word's hypernym. Classifying co-hyponyms as incompatibles and non-cohyponyms as compatibles performs very poorly (7 and 18 F1-scores for compatibility and incompatibility, respectively). On the other hand, the opposite strategy – co-hyponyms as compatibles and non-cohyponyms as incompatibles – works much better (62 and 84 F1), even outperforming many supervised models. Yet, this strategy does not suffice. For example, all animal pairs would be treated as compatibles, whereas 54% of them are actually incompatible. By contrast the L2 interaction model gets 78% of these incompatible pairs right.

5 Conclusion

We have introduced the challenge of modeling compatibility to the computational linguistics community. To this end, we collected a data set, and produced a model that satisfactorily captures a large portion of the data, that cannot be accounted for by simple semantic relatedness. Finally, we have explored the features learned by the model, confirming that high-order category information is relevant for producing compatibility judgements.

Computational models of compatibility could help in many semantic tasks, such as coreference resolution, question answering, modeling plausibility and negation. Future lines of research will explore the contributions that accounting for compatibility can make to these tasks.

Acknowledgments

We thank Denis Paperno for the interesting discussions that motivated this paper and the three anonymous reviewers for useful comments. We acknowledge ERC 2011 Starting Independent Research Grant n. 283554 (COMPOSES).

References

Heike Adel and Hinrich Schütze. 2014. Using mined coreference chains as a resource for a semantic task. In *Proceedings of EMNLP*, pages 1447–1452, Doha, Qatar.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of ACL*, pages 238–247, Baltimore, MD.

Stephen Clark. 2015. Vector space models of lexical meaning. In Shalom Lappin and Chris Fox, editors, *Handbook of Contemporary Semantics, 2nd ed.* Blackwell, Malden, MA. In press; `http://www.cl.cam.ac.uk/~sc609/pubs/sem_handbook.pdf`.

Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. 2011. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*.

D. Alan Cruse. 1986. *Lexical Semantics*. Cambridge University Press, Cambridge, UK.

Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.

Karl Moritz Hermann, Edward Grefenstette, and Phil Blunsom. 2013. "Not not bad" is not "bad": A distributional account of negation. In *Proceedings of ACL Workshop on Continuous Vector Space Models and their Compositionality*, pages 74–82, Sofia, Bulgaria.

Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 2012. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, Berlin.

Dermot Lynott and Louise Connell. 2009. Embodied conceptual combination. *Frontiers in Psychology*, 1:212.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. `http://arxiv.org/abs/1301.3781/`.

Saif Mohammad, Bonnie Dorr, Graeme Hirst, and Peter Turney. 2013. Computing lexical contrast. *Computational Linguistics*, 39(3):555–590.

M. Lynne Murphy. 2010. Antonymy and incompatibility. In Keith Allan, editor, *Concise Encyclopedia of Semantics*. Elsevier, Amsterdam.

Massimo Poesio, Simone Ponzetto, and Yannick Versley. 2010. Computational models of anaphora resolution: A survey. `http://clic.cimec.unitn.it/massimo/Publications/lilt.pdf`.

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

Eva Maria Vecchi, Marco Baroni, and Roberto Zamparelli. 2011. (Linear) maps of the impossible: Capturing semantic anomalies in distributional space. In *Proceedings of the ACL Workshop on Distributional Semantics and Compositionality*, pages 1–9, Portland, OR.