Learning Adjective Meanings with a Tensor-Based Skip-Gram Model

Jean Maillard
University of Cambridge
Computer Laboratory
jean@maillard.it

Stephen Clark
University of Cambridge
Computer Laboratory
sc609@cam.ac.uk

Abstract

We present a compositional distributional semantic model which is an implementation of the tensor-based framework of Coecke et al. (2011). It is an extended skip-gram model (Mikolov et al., 2013) which we apply to adjective-noun combinations, learning nouns as vectors and adjectives as matrices. We also propose a novel measure of adjective similarity, and show that adjective matrix representations lead to improved performance in adjective and adjective-noun similarity tasks, as well as in the detection of semantically anomalous adjective-noun pairs.

1 Introduction

A number of approaches have emerged for combining compositional and distributional semantics. Some approaches assume that all words and phrases are represented by vectors living in the same semantic space, and use mathematical operations such as vector addition and element-wise multiplication to combine the constituent vectors (Mitchell and Lapata, 2008). In these relatively simple methods, the composition function does not typically depend on its arguments or their syntactic role in the sentence.

An alternative which makes more use of grammatical structure is the recursive neural network approach of Socher et al. (2010). Constituent vectors in a phrase are combined using a matrix and non-linearity, with the resulting vector living in the same vector space as the inputs. The matrices can be parameterised by the syntactic type of the combining words or phrases (Socher et al., 2013; Hermann and Blunsom, 2013). Socher et al. (2012) extend this idea by representing the meanings of words and phrases as both a vector and a matrix, introducing a form of lexicalisation into the model.

A further extension, which moves us closer to formal semantics (Dowty et al., 1981), is to build a semantic representation in step with the syntactic derivation, and have the embeddings of words be determined by their syntactic type. Coecke et al. (2011) achieve this by treating relational words such as verbs and adjectives as functions in the semantic space. The functions are assumed to be multilinear maps, and are therefore realised as tensors, with composition being achieved through tensor contraction.[1] While the framework specifies the "shape" or semantic type of these tensors, it makes no assumption about how the values of these tensors should be interpreted (nor how they can be learned).

A proposal for the case of adjective-noun combinations is given by Baroni and Zamparelli (2010) (and also Guevara (2010)). Their model represents adjectives as matrices over noun space, trained via linear regression to approximate the "holistic" adjective-noun vectors from the corpus.

In this paper we propose a new solution to the problem of learning adjective meaning representations. The model is an implementation of the tensor framework of Coecke et al. (2011), here applied to adjective-noun combinations as a starting point. Like Baroni and Zamparelli (2010), our model also learn nouns as vectors and adjectives as matrices, but uses a skip-gram approach with negative sampling (Mikolov et al., 2013), extended to learn matrices.

We also propose a new way of quantifying adjective similarity, based on the action of adjectives on nouns (consistent with the view that adjectives are functions). We use this new measure instead of the naive cosine similarity function applied to matrices, and obtain competitive performance compared to the baseline skip-gram vectors (Mikolov et al., 2013) on an adjective similarity task. We also perform competitively on the

[1] Baroni et al. (2014) have developed a similar approach.

Proceedings of the 19th Conference on Computational Language Learning, pages 327–331,
Beijing, China, July 30-31, 2015.

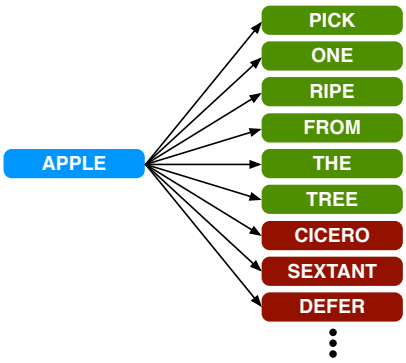

Figure 1: Learning the vector for *apple* in the context *pick one ripe apple from the tree*. The vector for *apple* is updated in order to increase the inner product with green vectors and decrease it with red ones, which are negatively sampled.

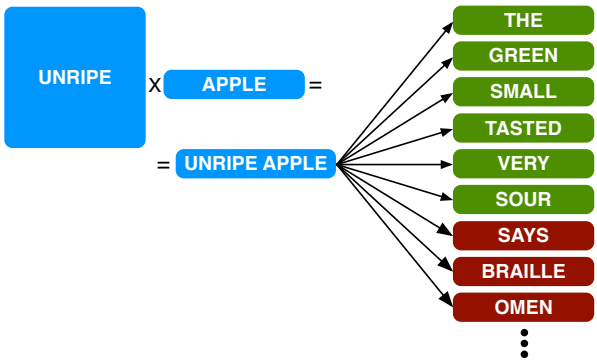

Figure 2: Learning the matrix for *unripe* in the context *the green small unripe apple tasted very sour*. The matrix for *unripe* is updated to increase the inner product of the vector for *unripe apple* with green vectors and decrease it with red ones.

adjective-noun similarity dataset from Mitchell and Lapata (2010). Finally, the tensor-based skip-gram model also leads to improved performance in the detection of semantically anomalous adjective-noun phrases, compared to previous work.

2 A tensor-based skip-gram model

Our model treats adjectives as linear maps over the vector space of noun meanings, encoded as matrices. The algorithm works in two stages: the first stage learns the noun vectors, as in a standard skip-gram model, and the second stage learns the adjective matrices, given fixed noun vectors.

2.1 Training of nouns

To learn noun vectors, we use a skip-gram model with negative sampling (Mikolov et al., 2013). Each noun n in the vocabulary is assigned two d-dimensional vectors: a *content* vector $\boldsymbol{n}$, which constitutes the embedding, and a *context* vector $\boldsymbol{n}'$. For every occurrence of a noun n in the corpus, the embeddings are updated in order to maximise the objective function

$$\sum_{\boldsymbol{c}' \in \mathcal{C}} \log \sigma(\boldsymbol{n} \cdot \boldsymbol{c}') + \sum_{\overline{\boldsymbol{c}}' \in \overline{\mathcal{C}}} \log \sigma(-\boldsymbol{n} \cdot \overline{\boldsymbol{c}}'), \quad (1)$$

where $\mathcal{C}$ is a set of contexts for the current noun, and $\overline{\mathcal{C}}$ is a set of negative contexts. The contexts are taken to be the vectors of words in a fixed window around the noun, while the negative contexts are vectors for k words sampled from a unigram distribution raised to the power of $3/4$ (Goldberg, 2014). In our experiments, we have set $k = 5$.

After each step, both content and context vectors are updated via back-propagation. This procedure leads to noun embeddings (content vectors) which have a high inner product with the vectors of words in the context of the noun, and a low inner product with vectors of negatively sampled words. Fig. 1 shows this intuition.

2.2 Training of adjectives

Each adjective a in the vocabulary is assigned a matrix $\mathbf{A}$, initialised to the identity plus uniform noise. First, all adjective-noun phrases (a, n) are extracted from the corpus. For each (a, n) pair, the corresponding adjective matrix $\mathbf{A}$ and noun vector $\boldsymbol{n}$ are multiplied to compute the adjective-noun vector $\mathbf{A}\boldsymbol{n}$. The matrix $\mathbf{A}$ is then updated to maximise the objective function

$$\sum_{\boldsymbol{c}' \in \mathcal{C}} \log \sigma(\mathbf{A}\boldsymbol{n} \cdot \boldsymbol{c}') + \sum_{\overline{\boldsymbol{c}}' \in \overline{\mathcal{C}}} \log \sigma(-\mathbf{A}\boldsymbol{n} \cdot \overline{\boldsymbol{c}}'). \quad (2)$$

The contexts $\mathcal{C}$ are taken to be the vectors of words in a window around the adjective-noun phrase, while the negative contexts $\overline{\mathcal{C}}$ are again vectors of randomly sampled words. Matrices are initialised to the identity, while the context vectors $\mathcal{C}$ are the results of Section 2.1.

Finally, the matrix $\mathbf{A}$ is updated via back-propagation. Equation 2 means that the induced matrices will have the following property: when multiplying the matrix with a noun vector, the resulting adjective-noun vector will have a high inner product with words in the context window of the adjective-noun phrase, and low inner product for negatively sampled words. This is exemplified in Figure 2.

2.3 Similarity measure

The similarity of two vectors $\boldsymbol{n}$ and $\boldsymbol{m}$ is generally measured using the cosine similarity function (Turney and Pantel, 2010; Baroni et al., 2014),

$$\text{vecsim}(\boldsymbol{n}, \boldsymbol{m}) = \frac{\boldsymbol{n} \cdot \boldsymbol{m}}{\|\boldsymbol{n}\| \, \|\boldsymbol{m}\|}.$$

Based on tests using a development set, we found that using cosine to measure the similarity of adjective matrices leads to no correlation with gold-standard similarity judgements. Cosine similarity, while suitable for vectors, does not capture any information about the function of matrices as linear maps. We postulate that a suitable measure of the similarity of two adjectives should be related to how similarly they transform nouns.

Consider two adjective matrices $\mathbf{A}$ and $\mathbf{B}$. If $\mathbf{A}\boldsymbol{n}$ and $\mathbf{B}\boldsymbol{n}$ are similar vectors for every noun vector $\boldsymbol{n}$, then we deem the adjectives to be similar. Therefore, one possible measure involves calculating the cosine distance between the images of all nouns under the two adjectives, and taking the average or median of these distances. Rather than working on every noun in the vocabulary, which is expensive, we instead take the most frequent nouns, cluster them, and use the cluster centroids (obtained in our case using k-means). The resulting distance function is given by

$$\text{matsim}(\mathbf{A}, \mathbf{B}) = \underset{\boldsymbol{n} \in \mathcal{N}}{\text{median}} \ \text{vecsim}(\mathbf{A}\boldsymbol{n}, \mathbf{B}\boldsymbol{n}), \tag{3}$$

where the median is taken over the set of cluster centroids $\mathcal{N}$.[2]

3 Evaluation

The model is trained on a dump of the English Wikipedia, automatically parsed with the C&C parser (Clark and Curran, 2007). The corpus contains around 200 million noun examples, and 30 million adjective-noun examples. For every context word in the corpus, 5 negative words are sampled from the unigram distribution. Subsampling is used to decrease the number of frequent words (Mikolov et al., 2013). We train 100-dimensional noun vectors and 100×100-dimensional adjective matrices.

3.1 Word Similarity

First we test word, rather than phrase, similarity on the MEN test collection (Bruni et al., 2014), which contains a set of POS-tagged word pairs together with gold-standard human similarity judgements. We use the POS tags to select all noun-noun and adjective-adjective pairs, leaving us with a set of 643 noun-noun pairs and 96 adjective-adjective pairs. For the noun-noun dataset, we are testing the quality of the 100-dimensional noun vectors from the first stage of the tensor-based skip-gram model (TBSG), which is essentially `word2vec` applied to just learning noun vectors. These are compared to the 300-dimensional SKIPGRAM vectors available from the `word2vec` page (which have been trained on a very large news corpus).[3]

MODEL	CORRELATION
SKIPGRAM-300	**0.776**
TBSG-100	0.769

Table 1: Spearman rank correlation on noun similarity task.

MODEL	CORRELATION
TBSG-100×100	**0.645**
SKIPGRAM-300	0.638

Table 2: Spearman rank correlation on adjective similarity task.

The adjective-adjective pairs are used to test the 100×100 matrices obtained from our TBSG model, again compared to the 300-dimensional SKIPGRAM vectors. The Spearman correlations between human judgements and the similarity of vectors are reported in Tables 1 and 2. Note that for adjectives we used the similarity measure described in Section 2.3. Table 1 shows that the noun vectors we use are of a high quality, performing comparably to the SKIPGRAM noun vectors on the noun-noun similarity data. Table 2 shows our TBSG adjective matrices, plus new similarity measure, to also perform comparably to the SKIPGRAM adjective vectors on the adjective-adjective similarity data.

3.2 Phrase Similarity

The TBSG model aims to learn matrices that act in a compositional manner. Therefore, a more interesting evaluation of its performance is to test how well the matrices combine with noun vectors.

[2] We chose the median instead of the average as it is more resistant to outliers in the data.

[3] `http://word2vec.googlecode.com/`

MODEL	CORRELATION
TBSG-100	**0.50**
SKIPGRAM-300 (add)	0.48
SKIPGRAM-300 (N only)	0.43
TBSG-100 (N only)	0.42
REG-600	0.37
humans	*0.52*

Table 3: Spearman rank correlation on adjective-noun similarity task.

	COSINE		DENSITY	
MODEL	t	sig.	t	sig.
TBSG-100	**5.16**	$***$	**5.72**	$***$
ADD-300	0.31		2.63	$**$
MUL-300	-0.56		2.68	$**$
REG-300	0.48		3.12	$**$

Table 4: Correlation on test data for semantic anomalies. Significance levels are marked $***$ for $p < 0.001$, $**$ for $p < 0.01$.

We use the Mitchell and Lapata (2010) adjective-noun similarity dataset, which contains pairs of adjective-noun phrases such as *last number – vast majority* together with gold-standard human similarity judgements. For the evaluation, we calculate the Spearman correlation between non-averaged human similarity judgements and the cosine similarity of the vectors produced by various compositional models.

The results in Table 3 show that TBSG has the best correlation with human judgements of the other models tested. It outperforms SKIPGRAM vectors with both addition and element-wise multiplication as composition functions (the latter not shown in that table, as it is worse than addition). Also reported is the baseline performance of SKIPGRAM and TBSG when using only nouns to compute similarity (ignoring the adjectives). It is interesting to note that TBSG also outperforms the result of the matrix-vector linear regression method (REG-600) of Baroni and Zamparelli (2010) as reported by Vecchi et al. (2015) on the same dataset. Their method trains a matrix for every adjective via linear regression to approximate corpus-extracted "holistic" adjective-noun vectors, and is therefore similar in spirit to TBSG.

3.3 Semantic Anomaly

Finally, we use the model to distinguish between semantically acceptable and anomalous adjective-noun phrases, using the data from Vecchi et al. (2011). The data consists of two sets: a set of unobserved acceptable phrases (e.g. *ethical statute*) and one of deviant phrases (e.g. *cultural acne*). Following Vecchi et al. (2011) we use two indices of semantic anomaly. The first, denoted COSINE, is the cosine between the adjective-noun vector and the noun vector. This is based on the hypothesis that deviant adjective-noun vectors will form a wider angle with the noun vector. The second index, denoted DENSITY, is the average cosine distance between the adjective-noun vector and its 10 nearest noun neighbours. This measure is based on the hypothesis that nonsensical adjective-nouns should not have many neighbours in the space of (meaningful) nouns.[4] These two measures are computed for the acceptable and deviant sets, and compared using a two-tailed Welch's t-test.

Table 4 compares the performance of TBSG with the results of count-based vectors using addition (ADD) and element-wise multiplication (MUL) reported by Vecchi et al. (2011), as well as the matrix-vector linear regression method (REG-300) of Baroni and Zamparelli (2010). TBSG obtains the highest scores with both measures.

4 Conclusions

In this paper we have implemented the tensor-based framework of Coecke et al. (2011) in the form of a skip-gram model extended to learn higher-order embeddings, in this case adjectives as matrices. While adjectives and nouns are learned separately in this study, an obvious extension is to learn embeddings jointly. We find the tensor-based skip-gram model particularly attractive for the obvious ways in which it can be extended to other parts-of-speech (Maillard et al., 2014). For example, in this framework transitive verbs are third-order tensors which yield a sentence vector when contracted with subject and object vectors. Assuming contextual representations of sentences, these could be learned by the tensor-based skip-gram as a straightforward extension from second-order (matrices) to third-order tensors (and potentially beyond for words requiring even higher order tensors).

[4] Vecchi et al. (2011) also use a third index of semantic anomaly, based on the length of adjective-noun vectors. We omit this measure as we deem it unsuitable for models not based on context counts and elementwise vector operations.

Acknowledgements

Jean Maillard is supported by an EPSRC Doctoral Training Grant and a St John's Scholarship. Stephen Clark is supported by ERC Starting Grant DisCoTex (306920) and EPSRC grant EP/I037512/1. We would like to thank Tamara Polajnar, Laura Rimell, and Eva Vecchi for useful discussion.

References

Marco Baroni, Roberto Zamparelli 2010. *Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space*. *Proceedings of EMNLP*, 1183–1193.

Marco Baroni, Raffaela Bernardi, Roberto Zamparelli 2014 *Frege in space: A program for compositional distributional semantics*. *Linguistic Issues in Language Technologies* 9(6), 5–110.

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

Elia Bruni, Nam Khanh Tran, Marco Baroni 2014 *Multimodal Distributional Semantics*. *Journal of Artificial Intelligence Research* 49, 1–47.

Stephen Clark, James R. Curran 2007 *Wide-coverage efficient statistical parsing with CCG and log-linear models*. *Computational Linguistics* 33(4), 493–552.

Bob Coecke, Mehrnoosh Sadrzadeh, Stephen Clark 2011. *Mathematical Foundations for a Compositional Distributional Model of Meaning*. *Linguistic Analysis* 36 (Lambek Festschrift), 345–384.

David R. Dowty, Robert E. Wall, Stanley Peters 1981 *Introduction to Montague semantics*. *Dordrecht*.

Yoav Goldberg, Omer Levy 2014 *word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method*. *arXiv preprint* 1402.3722.

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

Karl Moritz Hermann and Phil Blunsom. 2013 *The role of syntax in vector space models of compositional semantics*. *Proceedings of ACL*, 894–904.

Jean Maillard, Stephen Clark, Edward Grefenstette 2014. *A Type-Driven Tensor-Based Semantics for CCG*. *Proceedings of the EACL 2014 Type Theory and Natural Language Semantics Workshop*, 46–54.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeff Dean 2013. *Distributed representations of words and phrases and their compositionality*. *Proceedings of NIPS*, 3111–3119.

Jeff Mitchell, Mirella Lapata 2008 *Vector-based models of semantic composition*. *Proceedings of ACL 08*, 263–244.

Jeff Mitchell, Mirella Lapata 2010 *Composition in Distributional Models of Semantics*. *Cognitive science* 34(8), 1388–1439.

Richard Socher, Christopher D. Manning, Andrew Y. Ng 2010 *Learning continuous phrase representations and syntactic parsing with recursive neural networks*. *Proceedings of the NIPS Deep Learning and Unsupervised Feature Learning Workshop*, 1–9.

Richard Socher, Brody Huval, Christopher D. Manning, Andrew Y. Ng 2012 *Semantic compositionality through recursive matrix-vector spaces*. *Proceedings of EMNLP*, 1201–1211.

Richard Socher, John Bauer, Christopher D. Manning, Andrew Y. Ng 2013 *Parsing with Compositional Vector Grammars*. *Proceedings of ACL*, 455-465.

Mark Steedman 2000 *The Syntactic Process*. *MIT Press*.

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

Eva M. Vecchi, Marco Baroni, Roberto Zamparelli 2011 *(Linear) maps of the impossible: Capturing semantic anomalies in distributional space*. *Proceedings of ACL DISCo Workshop*, 1–9.

Eva M. Vecchi, Marco Marelli, Roberto Zamparelli, Marco Baroni 2015 *Spicy adjectives and nominal donkeys: Capturing semantic deviance using compositionality in distributional spaces*. *Accepted for publication*.