Jointly optimizing word representations for lexical and sentential tasks with the C-PHRASE model

Nghia The Pham Germán Kruszewski Angeliki Lazaridou Marco Baroni
Center for Mind/Brain Sciences
University of Trento
{thenghia.pham|german.kruszewski|angeliki.lazaridou|marco.baroni}@unitn.it

Abstract

We introduce C-PHRASE, a distributional semantic model that learns word representations by optimizing context prediction for phrases at all levels in a syntactic tree, from single words to full sentences. C-PHRASE outperforms the state-of-the-art C-BOW model on a variety of lexical tasks. Moreover, since C-PHRASE word vectors are induced through a compositional learning objective (modeling the contexts of words combined into phrases), when they are summed, they produce sentence representations that rival those generated by *ad-hoc* compositional models.

1 Introduction

Distributional semantic models, that induce vector-based meaning representations from patterns of co-occurrence of words in corpora, have proven very successful at modeling many lexical relations, such as synonymy, co-hyponomy and analogy (Mikolov et al., 2013c; Turney and Pantel, 2010). The recent evaluation of Baroni et al. (2014b) suggests that the C-BOW model introduced by Mikolov et al. (2013a) is, consistently, the best across many tasks.[1]

Interestingly, C-BOW vectors are estimated with a simple compositional approach: The weights of adjacent words are jointly optimized so that their sum will predict the distribution of their contexts. This is reminiscent of how the parameters of some *compositional* distributional semantic models are estimated by optimizing the prediction of the contexts in which phrases occur in corpora (Baroni and Zamparelli, 2010; Guevara, 2010; Dinu et al., 2013). However, these compositional approaches assume that word vectors have already been constructed, and contextual evidence is only used to induce optimal combination rules to derive representations of phrases and sentences.

In this paper, we follow through on this observation to propose the new C-PHRASE model. Similarly to C-BOW, C-PHRASE learns word representations by optimizing their joint context prediction. However, unlike in flat, window-based C-BOW, C-PHRASE groups words according to their syntactic structure, and it simultaneously optimizes context-predictions at different levels of the syntactic hierarchy. For example, given training sentence *"A sad dog is howling in the park"*, C-PHRASE will optimize context prediction for *dog*, *sad dog*, *a sad dog*, *a sad dog is howling*, etc., but not, for example, for *howling in*, as these two words do not form a syntactic constituent by themselves.

C-PHRASE word representations outperform C-BOW on several word-level benchmarks. In addition, because they are estimated in a compositional way, C-PHRASE word vectors, when combined through simple addition, produce sentence representations that are better than those obtained when adding other kinds of vectors, and competitive against *ad-hoc* compositional methods on various sentence meaning benchmarks.

2 The C-PHRASE model

We start with a brief overview of the models proposed by Mikolov et al. (2013a), as C-PHRASE builds on them. The **Skip-gram** model derives the vector of a target word by setting its weights to predict the words surrounding it in the corpus.

[1] We refer here not only to the results reported in Baroni et al. (2014b), but also to the more extensive evaluation that Baroni and colleagues present in the companion website (http://clic.cimec.unitn.it/composes/semantic-vectors.html). The experiments there suggest that only the Glove vectors of Pennington et al. (2014) are competitive with C-BOW, and only when trained on a corpus several orders of magnitude larger than the one used for C-BOW.

More specifically, the objective function is:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \quad (1)$$

where the word sequence $w_1, w_2, ..., w_T$ is the training corpus and c is the size of the window around the target word w_t, consisting of the context words w_{t+j} that must be predicted by the induced vector representation for the target.

While Skip-gram learns each word representation separately, the **C-BOW** model takes their combination into account. More precisely, it tries to predict a context word from the combination of the previous and following words, where the combination method is vector addition. The objective function is:

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_t|w_{t-c}..w_{t-1}, w_{t+1}..w_{t+c}) \quad (2)$$

While other distributional models consider sequences of words jointly as *context* when estimating the parameters for a single word (Agirre et al., 2009; Melamud et al., 2014), C-BOW is unique in that it estimates the weights of a sequence of words jointly, based on their shared context. In this respect, C-BOW extends the distributional hypothesis (Harris, 1954) that words with similar context distributions should have similar meanings to longer sequences. However, the word combinations of C-BOW are not natural linguistic constituents, but arbitrary n-grams (e.g., sequences of 5 words with a gap in the middle). Moreover, the model does not attempt to capture the *hierarchical* nature of syntactic phrasing, such that *big brown dog* is a meaningful phrase, but so are its children *brown dog* and *dog*.

C-PHRASE aims at capturing the same intuition that word combinations with similar context distributions will have similar meaning, but it applies it to syntactically motivated, potentially nested phrases. More precisely, we estimate word vectors such that they and their summed combinations are able to predict the contexts of words, phrases and sentences. The model is formalized as follows. We start from a parsed text corpus $\mathbb{T}$, composed of constituents $\mathcal{C}[w_l, \cdots, w_r]$, where $w_l, \cdots, w_r$ are the words spanned by the constituent, located in positions l to r in the corpus. We minimize an objective function analogous to equations (1) and (2), but instead of just using individual words or bags of words to predict context, we use summed vector representations of well-formed constituents at all levels in the syntactic tree to predict the context of these constituents. There are similarities with both CBOW and Skip-gram. At the leaf nodes, C-PHRASE acts like Skip-gram, whereas at higher node in the parse tree, it behaves like CBOW model. Concretely, we try to predict the words located within a window $c_{\mathcal{C}}$ from every constituent in the parse tree.[2] In order to do so, we learn vector representations for words v_w by maximizing the sum of the log probabilities of the words in the context window of the well-formed constituents with stochastic gradient descent:

$$\sum_{\mathcal{C}[w_l,\cdots,w_r]\in\mathbb{T}}\sum_{1\leq j\leq c_{\mathcal{C}}}\Big(\log p(w_{l-j}|\mathcal{C}[w_l,\cdots,w_r]) + \log p(w_{r+j}|\mathcal{C}[w_l,\cdots,w_r])\Big) \quad (3)$$

with p theoretically defined as:

$$p(w_O|\mathcal{C}[w_l,\cdots,w_r]) = \frac{\exp\left(v'^{\top}_{w_O}\frac{\sum_{i=l}^{r}v_{w_i}}{r-l+1}\right)}{\sum_{w=1}^{W}\exp\left(v'^{\top}_{w}\frac{\sum_{i=l}^{r}v_{w_i}}{r-l+1}\right)}$$

where W is the size of the vocabulary, v' and v denote output (context) and input vectors, respectively, and we take the input vectors to represent the words. In practice, since the normalization constant for the above probability is expensive to compute, we follow Mikolov et al. (2013b) and use negative sampling.

We let the context window size $c_{\mathcal{C}}$ vary as a function of the height of the constituent in the syntactic tree. The height $h(\mathcal{C})$ of a constituent is given by the maximum number of intermediate nodes separating it from any of the words it dominates (such that $h = 0$ for words, $h = 1$ for two-word phrases, etc.). Then, for a constituent of height $h(\mathcal{C})$, C-PHRASE considers $c_{\mathcal{C}} = c_1 + h(\mathcal{C})c_2$ context words to its left and right (the non-negative integers c_1 and c_2 are hyperparameters of the model; with $c_2 = 0$, context becomes constant

[2] Although here we only use single words as context, the latter can be extended to encompass any sensible linguistic item, e.g., frequent n-grams or, as discussed below, syntactically-mediated expressions

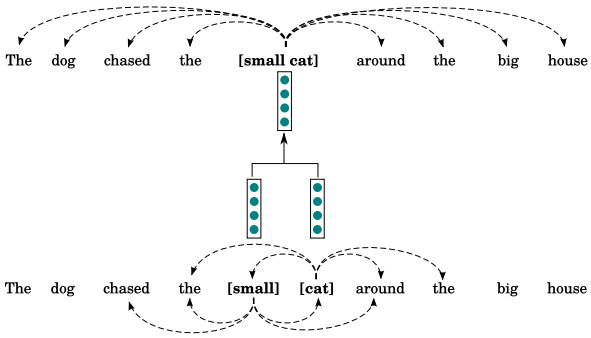

Figure 1: C-PHRASE context prediction objective for the phrase *small cat* and its children. The phrase vector is obtained by summing the word vectors. The predicted window is wider for the higher constituent (the phrase).

across heights). The intuition for enlarging the window proportionally to height is that, for shorter phrases, narrower contexts are likely to be most informative (e.g., a modifying adjective for a noun), whereas for longer phrases and sentences it might be better to focus on broader "topical" information spread across larger windows (paragraphs containing sentences about weather might also contain the words *rain* and *sun*, but without any tendency for these words to be perfectly adjacent to the target sentences).

Figure 1 illustrates the prediction objective for a two-word phrase and its children. Since all constituents (except the topmost) form parts of larger constituents, their representations will be learned both from the objective of predicting their own contexts, and from error propagation from the same objective applied higher in the tree. As a side effect, words, being lower in the syntactic tree, will have their vectors updated more often, and thus might have a greater impact on the learned parameters. This is another reason for varying window size with height, so that the latter effect will be counter-balanced by higher constituents having larger context windows to predict.

For lexical tasks, we directly use the vectors induced by C-PHRASE as word representations. For sentential tasks, we simply add the vectors of the words in a sentence to obtain its representation, exploiting the fact that C-PHRASE was trained to predict phrase contexts from the additive combination of their elements.

Joint optimization of word and phrase vectors The C-PHRASE hierarchical learning objective can capture, in parallel, generalizations about the contexts of words and phrases at different levels of complexity. This results, as we will see, in better word vectors, presumably because C-PHRASE is trained to predict how the contexts of a word change based on its phrasal collocates (*cup* will have very different contexts in *world cup* vs. *coffee cup*). At the same time, because the vectors are optimized based on their occurrence in phrases of different syntactic complexity, they produce good sentence representations when they are combined. To the best of our knowledge, C-PHRASE is the first model that is jointly optimized for lexical and compositional tasks. C-BOW uses shallow composition information to learn word vectors. Conversely, some compositional models –e.g., Kalchbrenner et al. (2014), Socher et al. (2013)– induce word representations, that are only optimized for a compositional task and are not tested at the lexical level. Somewhat relatedly to what we do, Hill et al. (2014) evaluated representations learned in a sentence translation task on word-level benchmarks. Some *a priori* justification for treating word and sentence learning as joint problems comes from human language acquisition, as it is obvious that children learn word and phrase meanings in parallel and interactively, not sequentially (Tomasello, 2003).

Knowledge-leanness and simplicity For training, C-PHRASE requires a large, syntactically-parsed corpus (more precisely, it only requires the constituent structure assigned by the parser, as it is blind to syntactic labels). Both large unannotated corpora and efficient pre-trained parsers are available for many languages, making the C-PHRASE knowledge demands feasible for practical purposes. There is no need to parse the sentences we want to build representations for at test time, since the component word vectors are simply added. The only parameters of the model are the word vectors; specifically, no extra parameters are needed for composition (composition models such as the one presented in Socher et al. (2012) require an extra parameter matrix for each word in the vocabulary, and even leaner models such as the one of Guevara (2010) must estimate a parameter matrix for each composition rule in the grammar). This makes C-PHRASE as simple as additive and multiplicative composition (Mitchell and

Lapata, 2010),[3] but C-PHRASE is both more effective in compositional tasks (see evaluation below), and it has the further advantage that it learns its own word vectors, thus reducing the number of arbitrary choices to be made in modeling.

Supervision Unlike many recent composition models (Kalchbrenner and Blunsom, 2013; Kalchbrenner et al., 2014; Socher et al., 2012; Socher et al., 2013, among others), the context-prediction objective of C-PHRASE does not require annotated data, and it is meant to provide general-purpose representations that can serve in different tasks. C-PHRASE vectors can also be used as initialization parameters for fully supervised, task-specific systems. Alternatively, the current unsupervised objective could be combined with task-specific supervised objectives to fine-tune C-PHRASE to specific purposes.

Sensitivity to syntactic structure During training, C-PHRASE is sensitive to syntactic structure. To cite an extreme example, *boy flowers* will be joined in a context-predicting phrase in *"these are considered [boy flowers]"*, but not in *"he gave [the boy] [flowers]"*. A more common case is that of determiners, that will only occur in phrases that also contain the following word, but not necessarily the preceding one. Sentence composition at test time, on the other hand, is additive, and thus syntax-insensitive. Still, the vectors being combined will reflect syntactic generalizations learned in training. Even if C-PHRASE produces the same representation for *red*+*car* and *car*+*red*, this representation combines a *red* vector that, during training, has often occurred in the modifier position of adjective-noun phrases, whereas *car* will have often occurred in the corresponding head position. So, presumably, the *red*+*car*=*car*+*red* vector will encode the adjective-noun asymmetry induced in learning. While the model won't be able to distinguish the rare cases in which *car red* is genuinely used as a phrase, in realistic scenarios this won't be a problem, because only *red car* will be encountered. In this respect, the successes and failures of C-PHRASE can tell us to what extent word order information is truly distinctive in practice, and to what extent it can instead be reconstructed from the typical role that words play in sentences.

[3] We do not report results for component-wise multiplicative in our evaluation because it performed much worse than addition in all the tasks.

Comparison with traditional syntax-sensitive word representations Syntax has often been exploited in distributional semantics for a richer characterization of *context*. By relying on a syntactic parse of the input corpus, a distributional model can take more informative contexts such as *subject-of-eat* vs. *object-of-eat* into account (Baroni and Lenci, 2010; Curran and Moens, 2002; Grefenstette, 1994; Erk and Padó, 2008; Levy and Goldberg, 2014a; Padó and Lapata, 2007; Rothenhäusler and Schütze, 2009). In this approach, syntactic information serves to select and/or enrich the *contexts* that are used to build representations of target units. On the other hand, we use syntax to determine the *target units* that we build representations for (in the sense that we jointly learn representations of their constituents). The focus is thus on unrelated aspects of model induction, and we could indeed use syntax-mediated contexts together with our phrasing strategy. Currently, given *eat (red apples)*, we treat *eat* as window-based context of *red apples*, but we could also take the context to be *object-of-eat*.

3 Evaluation

3.1 Data sets

Semantic relatedness of words In this classic lexical task, the models are required to quantify the degree of semantic similarity or relatedness of pairs of words in terms of cosines between the corresponding vectors. These scores are then compared to human gold standards. Performance is assessed by computing the correlation between system and human scores (Spearman correlation in all tasks except rg, where it is customary to report Pearson). We used, first of all, the MEN (**men**) data set of Bruni et al. (2014), that is split into 1K pairs for training/development, and 1K pairs for testing. We used the training set to tune the hyperparameters of our model, and report performance on the test set. The C-BOW model of Baroni et al. (2014b) achieved state-of-the art performance on MEN test. We also evaluate on the widely used WordSim353 set introduced by Finkelstein et al. (2002), which consists of 353 word pairs. The WordSim353 data were split by Agirre et al. (2009) into similarity (**wss**) and relatedness (**wsr**) subsets, focusing on strictly taxonomic (*television/radio*) vs. broader topical cases (*Maradona/football*), respectively. State-of-the-art performance on both sets is reported by Baroni

et al. (2014b), with the C-BOW model. We further consider the classic data set of Rubenstein and Goodenough (1965) (**rg**), consisting of 65 noun pairs. We report the state-of-the-art from Hassan and Mihalcea (2011), which exploited Wikipedia's linking structure.

Concept categorization Systems are asked to group a set of nominal concepts into broader categories (e.g. *arthritis* and *anthrax* into *illness*; *banana* and *grape* into *fruit*). As in previous work, we treat this as an unsupervised clustering task. We feed the similarity matrix produced by a model for all concepts in a test set to the CLUTO toolkit (Karypis, 2003), that clusters them into n groups, where n is the number of categories. We use standard CLUTO parameters from the literature, and quantify performance by cluster purity with respect to the gold categories. The Almuhareb-Poesio benchmark (Almuhareb, 2006) (**ap**) consists of 402 concepts belonging to 21 categories. A distributional model based on carefully chosen syntactic relations achieved top ap performance (Rothenhäusler and Schütze, 2009). The ESSLLI 2008 data set (Baroni et al., 2008) (**esslli**) consists of 6 categories and 42 concepts. State of the art was achieved by Katrenko and Adriaans (2008) by using full-Web queries and manually crafted patterns.

Semantic analogy The last lexical task we pick is analogy (**an**), introduced in Mikolov et al. (2013c). We focus on their *semantic* challenge, containing about 9K questions. In each question, the system is given a pair exemplifying a relation (*man*/*king*) and a test word (*woman*); it is then asked to find the word (*queen*) that instantiates the same relation with the test word as that of the example pair. Mikolov et al. (2013c) subtract the vector of the first word in a pair from the second, add the vector of the test word and look for the nearest neighbor of the resulting vector (e.g., find the word whose vector is closest to *king* - *man* + *woman*). We follow the method introduced by Levy and Goldberg (2014b), which returns the word x maximizing $\frac{\cos(king,x)\times\cos(woman,x)}{\cos(man,x)}$. This method yields better results for all models. Performance is measured by accuracy in retrieving the correct answer (in our search space of 180K words). The current state of the art on the semantic part and on the whole data set was reached by Pennington et al. (2014), who trained their word representations on a huge corpus consisting of 42B words.

Sentential semantic relatedness Similarly to word relatedness, composed sentence representations can be evaluated against benchmarks where humans provided relatedness/similarity scores for sentence pairs (sentences with high scores, such as *"A person in a black jacket is doing tricks on a motorbike"*/*"A man in a black jacket is doing tricks on a motorbike"* from the SICK data-set, tend to be near-paraphrases). Following previous work on these data sets, Pearson correlation is our figure of merit, and we report it between human scores and sentence vector cosine similarities computed by the models. SICK (Marelli et al., 2014) (**sick-r**) was created specifically for the purpose of evaluating compositional models, focusing on linguistic phenomena such as lexical variation and word order. Here we report performance of the systems on the test part of the data set, which contains 5K sentence pairs. The top performance (from the SICK SemEval shared task) was reached by Zhao et al. (2014) using a heterogeneous set of features that include WordNet and extra training corpora. Agirre et al. (2012) and Agirre et al. (2013) created two collections of sentential similarities consisting of subsets coming from different sources. From these, we pick the Microsoft Research video description dataset (**msrvid**), where near paraphrases are descriptions of the same short video, and the OnWN 2012 (**onwn1**) and OnWN 2013 (**onwn2**) data sets (each of these sets contains 750 pairs). The latter are quite different from other sentence relatedness benchmarks, since they compare definitions for the same or different words taken from WordNet and OntoNotes: these glosses often are syntactic fragments (*"cause something to pass or lead somewhere"*), rather than full sentences. We report top performance on these tasks from the respective shared challenges, as summarized by Agirre et al. (2012) and Agirre et al. (2013). Again, the top systems use feature-rich, supervised methods relying on distributional similarity as well as other sources, such as WordNet and named entity recognizers.

Sentential entailment Detecting the presence of entailment between sentences or longer passages is one of the most useful features that the computational analysis of text could provide (Dagan et al., 2009). We test our model on the SICK

entailment task (**sick-e**) (Marelli et al., 2014). All SICK sentence pairs are labeled as ENTAILING (*"Two teams are competing in a football match"/"Two groups of people are playing football"*), CONTRADICTING (*"The brown horse is near a red barrel at the rodeo"/"The brown horse is far from a red barrel at the rodeo"*) or NEUTRAL (*"A man in a black jacket is doing tricks on a motorbike"/"A person is riding the bicycle on one wheel"*). For each model, we train a simple SVM classifier based on 2 features: cosine similarity between the two sentence vectors, as given by the models, and whether the sentence pair contains a negation word (the latter has been shown to be a very informative feature for SICK entailment). The current state-of-the-art is reached by Lai and Hockenmaier (2014), using a much richer set of features, that include WordNet, the denotation graph of Young et al. (2014) and extra training data from other resources.

Sentiment analysis Finally, as sentiment analysis has emerged as a popular area of application for compositional models, we test our methods on the Stanford Sentiment Treebank (Socher et al., 2013) (**sst**), consisting of 11,855 sentences from movie reviews, using the coarse annotation into 2 sentiment degrees (*negative/positive*). We follow the official split into train (8,544), development (1,101) and test (2,210) parts. We train an SVM classifier on the training set, using the sentence vectors composed by a model as features, and report accuracy on the test set. State of the art is obtained by Le and Mikolov (2014) with the Paragraph Vector approach we describe below.

3.2 Model implementation

The source corpus we use to build the lexical vectors is created by concatenating three sources: ukWaC,[4] a mid-2009 dump of the English Wikipedia[5] and the British National Corpus[6] (about 2.8B words in total). We build vectors for the 180K words occurring at least 100 times in the corpus. Since our training procedure requires parsed trees, we parse the corpus using the Stanford parser (Klein and Manning, 2003).

C-PHRASE has two hyperparameters (see Section 2 above), namely basic window size (c_1) and height-dependent window enlargement factor (c_2). Moreover, following Mikolov et al. (2013b), during training we sub-sample less informative, very frequent words: this option is controlled by a parameter t, resulting in aggressive subsampling of words with relative frequency above it. We tune on MEN-train, obtaining $c_1 = 5$, $c_2 = 2$ and $t = 10^{-5}$. As already mentioned, sentence vectors are built by summing the vectors of the words in them.

In lexical tasks, we compare our model to the best C-BOW model from Baroni et al. (2014b),[7] and to a Skip-gram model built using the same hyperparameters as C-PHRASE (that also led to the best MEN-train results for Skip-gram).

In sentential tasks, we compare our model against adding the best C-BOW vectors pre-trained by Baroni and colleagues,[8] and adding our Skip-gram vectors. We compare the additive approaches to two sophisticated composition models. The first is the Practical Lexical Function (**PLF**) model of Paperno et al. (2014). This is a linguistically motivated model in the tradition of the "functional composition" approaches of Coecke et al. (2010) and Baroni et al. (2014a), and the only model in this line of research that has been shown to empirically scale up to real-life sentence challenges. In short, in the PLF model all words are represented by vectors. Words acting as argument-taking functions (such as verbs or adjectives) are also associated to one matrix for each argument they take (e.g., each transitive verb has a subject and an object matrix). Vector representations of arguments are recursively multiplied by function matrices, following the syntactic tree up to the top node. The final sentence representation is obtained by summing all the resulting vectors. The PLF approach requires syntactic parsing both in training and in testing and, more cumbersomely, to train a separate matrix for each argument slot of each function word (the training objective is again a context-predicting one). Here, we report PLF results on msrvid and onwn2 from Paperno et al. (2014), noting that they also used two simple but precious cues (word overlap and sentence length) we do not adopt here. We used their pre-trained vectors and matrices also for the SICK challenges, while the number of new ma-

[4] http://wacky.sslmit.unibo.it

[5] http://en.wikipedia.org

[6] http://www.natcorp.ox.ac.uk

[7] For fairness, we report their results when all tasks were evaluated with the same set of parameters, tuned on rg: this is row 8 of their Table 2.

[8] `http://clic.cimec.unitn.it/composes/semantic-vectors.html`

trices to estimate made it too time-consuming to implement this model in the onwn1 and sst tasks.

Finally, we test the Paragraph Vector (**PV**) approach recently proposed by Le and Mikolov (2014). Under PV, sentence representations are learned by predicting the words that occur in them. This unsupervised method has been shown by the authors to outperform much more sophisticated, supervised neural-network-based composition models on the sst task. We use our own implementation for this approach. Unlike in the original experiments, we found the PV-DBOW variant of PV to consistently outperform PV-DM, and so we report results obtained with the former.

Note that PV-DBOW aims mainly at providing representations for sentences, not words. When we do not need to induce vectors for sentences in the training corpus, i.e., only train to learn single words' representations and the softmax weights, PV-DBOW essentially reduces to Skip-gram. Therefore, we produce the PV-DBOW vectors for the sentences in the evaluation data sets using the softmax weights learned by Skip-gram. However, it is not clear that, if we were to train PV-DBOW jointly for words and sentences, we would get word vectors as good as those that Skip-gram induces.

4 Results

The results on the lexical tasks reported in Table 1 prove that C-PHRASE is providing excellent word representations, (nearly) as good or better than the C-BOW vectors of Baroni and colleagues in all cases, except for ap. Whenever C-PHRASE is not close to the state of the art results, the latter relied on richer knowledge sources and/or much larger corpora (ap, esslli, an).

Turning to the sentential tasks (Table 2), we first remark that using high-quality word vectors (such as C-BOW) and summing them leads to good results in all tasks, competitive with those obtained with more sophisticated composition models. This confirms the observation made by Blacoe and Lapata (2012) that simple-minded composition models are not necessarily worse than advanced approaches. Still, C-PHRASE is consistently better than C-BOW in all tasks, except sst, where the two models reach the same performance level.

C-PHRASE is outperforming PV on all tasks except sick-e, where the two models have the same performance, and onwn2, where PV is slightly better. C-PHRASE is outperforming PLF by a large margin on the SICK sets, whereas the two models are equal on msrvid, and PLF better on onwn2. Recall, however, that on the latter two benchmarks PLF used extra word overlap and sentence length features, so the comparison is not entirely fair.

The fact that state-of-the-art performance is well above our models is not surprising, since the SOA systems are invariably based on a wealth of knowledge sources, and highly optimized for a task. To put some of our results in a broader perspective, C-PHRASE's sick-r performance is 1% better than the median result of systems that participated in the SICK SemEval challenge, and comparable to that of Beltagy et al. (2014), who entered the competition with a system combining distributional semantics with a supervised probabilistic soft logic system. For sick-e (the entailment task), C-PHRASE's performance is less than one point below the median of the SemEval systems, and slightly above that of the Stanford submission, that used a recursive neural network with a tensor layer.

Finally, the performance of all our models, including PV, on sst is remarkably lower than the state-of-the-art performance of PV as reported by Le and Mikolov (2014). We believe that the crucial difference is that these authors estimated PV vectors *specifically on the sentiment treebank training data*, thus building *ad-hoc* vectors encoding the semantics of movie reviews. We leave it to further research to ascertain whether we could better fine-tune our models to sst by including the sentiment treebank training phrases in our source corpus.

Comparing vector lengths of C-BOW and C-PHRASE We gather some insight into how the C-PHRASE objective might adjust word representations for composition with respect to C-BOW by looking at how the length of word vectors changes across the two models.[9] While this is a very coarse measure, if a word vector is much longer/shorter (relative to the length of other word vectors of the same model) for C-PHRASE vs. C-BOW, it means that, when sentences are composed by addition, the effect of the word on the resulting sentence representation will be stronger/weaker.

[9] We performed the same analysis for C-PHRASE and Skip-gram, finding similar general trends to the ones we report for C-PHRASE and C-BOW.

	men	wss	wsr	rg	ap	esslli	an
Skip-gram	78	77	66	80	65	82	63
C-BOW	**80**	78	68	**83**	**71**	77	68
C-PHRASE	79	**79**	**70**	**83**	65	**84**	**69**
SOA	80	80	70	86	79	91	82

Table 1: Lexical task performance. See Section 3.1 for figures of merit (all in percentage form) and state-of-the-art references. C-BOW results (tuned on rg) are taken from Baroni et al. 2014b.

	sick-r	sick-e	msrvid	onwn1	onwn2	sst
Skip-gram	70	72	74	66	62	78
C-BOW	70	74	74	69	63	**79**
C-PHRASE	**72**	**75**	**79**	**70**	65	**79**
PLF	57	72	**79**	NA	**67**	NA
PV	67	**75**	77	66	66	77
SOA	83	85	88	71	75	88

Table 2: Sentential task performance. See Section 3.1 for figures of merit (all in percentage form) and state-of-the-art references. The PLF results on msrvid and onwn2 are taken from Paperno et al. 2014.

The relative-length-difference test returns the following words as the ones that are most severely de-emphasized by C-PHRASE compared to C-BOW: *be, that, an, not, they, he, who, when, well, have*. Clearly, C-PHRASE is weighting down grammatical terms that tend to be context-agnostic, and will be accompanied, in phrases, by more context-informative content words. Indeed, the list of terms that are instead emphasized by C-PHRASE include such content-rich, monosemous words as *gnawing, smackdown, demographics*. This is confirmed by a POS-level analysis that indicates that the categories that are, on average, most de-emphasized by C-PHRASE are: determiners, modals, pronouns, prepositions and (more surprisingly) proper nouns. The ones that are, in relative terms, more emphasized are: *-ing* verb forms, plural and singular nouns, adjectives and their superlatives. While this reliance on content words to the detriment of grammatical terms is not always good for sentential tasks (*"not always good"* means something very different from *"always good"*!), the convincing comparative performance of C-PHRASE in such tasks suggests that the semantic effect of grammatical terms is in any case beyond the scope of current corpus-based models, and often not crucial to attain competitive results on typical benchmarks (think, e.g., of how little modals, one of the categories that C-PHRASE downplays the most, will matter when detecting paraphrases that are based on picture descriptions).

We also applied the length difference test to words in specific categories, finding similar patterns. For example, looking at *-ly* adverbs only, those that are de-emphasized the most by C-PHRASE are *recently, eventually, originally, notably* and *currently* – all adverbs denoting temporal factors or speaker attitude. On the other hand, the ones that C-PHRASE lengthens the most, relative to C-BOW, are *clinically, divinely, ecologically, noisily* and *theatrically*: all adverbs with more specific, content-word-like meanings, that are better captured by distributional methods, and are likely to have a bigger impact on tasks such as paraphrasing or entailment.

Effects of joint optimization at word and phrase levels As we have argued before, C-PHRASE is able to obtain good word representations presumably because it learns to predict how the context of a word changes in the presence of different collocates. To gain further insight into this claim, we looked at the nearest neighbours of some example terms, like *neural*, *network* and *neural network* (the latter, composed by addition) both in C-PHRASE and C-BOW. The results for this particular example can be appreciated in Table 3.

Interestingly, while for C-BOW we observe some confusion between the meaning of the individual words and the phrase, C-PHRASE seems to provide more orthogonal representations for the lexical items. For example, *neural* in C-

C-BOW			C-PHRASE		
neural	*network*	*neural network*	*neural*	*network*	*neural network*
neuronal	networks	network	neuronal	networks	network
neurons	superjanet4	neural	cortical	internetwork	neural
hopfield	backhaul	networks	connectionist	wans	perceptron
cortical	fiber-optic	hopfield	neurophysiological	network.	networks
connectionist	point-to-multipoint	packet-switched	sensorimotor	multicasting	hebbian
feed-forward	nsfnet	small-world	sensorimotor	nsfnet	neurons
feedforward	multi-service	local-area	neocortex	networking	neocortex
neuron	circuit-switched	superjanet4	electrophysiological	tymnet	connectionist
backpropagation	wide-area	neuronal	neurobiological	x.25	neuronal

Table 3: Nearest neighbours of *neural*, *network* and *neural network* both for C-BOW and C-PHRASE

BOW contains neighbours that fit well with *neural network*, like *hopfield*, *connectionist* and *feed-forward*. Conversely, *neural network* has neighbours that correspond to *network* like *local-area* and *packet-switched*. In contrast, C-PHRASE neighbours for *neural* are mostly related to the *brain* sense of the word, e.g., *cortical*, *neurophysiological*, etc. (with the only exception of *connectionist*). The first neighbour of *neural network*, excluding its own component words, quite sensibly, is *perceptron*.

5 Conclusion

We introduced C-PHRASE, a distributional semantic model that is trained on the task of predicting the contexts surrounding phrases at all levels of a hierarchical sentence parse, from single words to full sentences. Consequently, word vectors are induced by taking into account not only their contexts, but also how co-occurrence with other words within a syntactic constituent is affecting these contexts.

C-PHRASE vectors outperform state-of-the-art C-BOW vectors in a wide range of lexical tasks. Moreover, because of the way they are induced, when C-PHRASE vectors are summed, they produce sentence representations that are as good or better than those obtained with sophisticated composition methods.

C-PHRASE is a very parsimonious approach: The only major resource required, compared to a completely knowledge-free, unsupervised method, is an automated parse of the training corpus (but no syntactic labels are required, nor parsing at test time). C-PHRASE has only 3 hyperparameters and no composition-specific parameter to tune and store.

Having established a strong empirical baseline with this parsimonious approach, in future research we want to investigate the impact of possible extensions on both lexical and sentential tasks. When combining the vectors, either for induction or composition, we will try replacing plain addition with other operations, starting with something as simple as learning scalar weights for different words in a phrase (Mitchell and Lapata, 2010). We also intend to explore more systematic ways to incorporate supervised signals into learning, to fine-tune C-PHRASE vectors to specific tasks.

On the testing side, we are fascinated by the good performance of additive models, that (at test time, at least) do not take word order nor syntactic structure into account. We plan to perform a systematic analysis of both existing benchmarks and natural corpus data, both to assess the actual impact that such factors have on the aspects of meaning we are interested in (take two sentences in an entailment relation: how often does shuffling the words in them make it impossible to detect entailment?), and to construct new benchmarks that are more challenging for additive methods.

The C-PHRASE vectors described in this paper are made publicly available at: `http://clic.cimec.unitn.it/composes/`.

Acknowledgments

We thank Gemma Boleda and the anonymous reviewers for useful comments. We acknowledge ERC 2011 Starting Independent Research Grant n. 283554 (COMPOSES).

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of HLT-NAACL*, pages 19–27, Boulder, CO.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: a

pilot on semantic textual similarity. In *Proceedings of *SEM*, pages 385–393, Montreal, Canada.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Proceedings of *SEM*, pages 32–43, Atlanta, GA.

Abdulrahman Almuhareb. 2006. *Attributes in Lexical Acquisition*. Phd thesis, University of Essex.

Marco Baroni and Alessandro Lenci. 2010. Distributional Memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

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.

Marco Baroni, Stefan Evert, and Alessandro Lenci, editors. 2008. *Bridging the Gap between Semantic Theory and Computational Simulations: Proceedings of the ESSLLI Workshop on Distributional Lexical Semantic*. FOLLI, Hamburg.

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

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

Islam Beltagy, Stephen Roller, Gemma Boleda, Katrin Erk, and Raymond Mooney. 2014. UTexas: Natural language semantics using distributional semantics and probabilistic logic. In *Proceedings of SemEval*, pages 796–801, Dublin, Ireland, August.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of EMNLP*, pages 546–556, Jeju Island, Korea.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36:345–384.

James Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In *Proceedings of the ACL Workshop on Unsupervised Lexical Acquisition*, pages 59–66, Philadelphia, PA.

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

Georgiana Dinu, Nghia The Pham, and Marco Baroni. 2013. General estimation and evaluation of compositional distributional semantic models. In *Proceedings of ACL Workshop on Continuous Vector Space Models and their Compositionality*, pages 50–58, Sofia, Bulgaria.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of EMNLP*, pages 897–906, Honolulu, HI.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer, Boston, MA.

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

Zellig Harris. 1954. Distributional structure. *Word*, 10(2-3):1456–1162.

Samer Hassan and Rada Mihalcea. 2011. Semantic relatedness using salient semantic analysis. In *Proceedings of AAAI*, pages 884–889, San Francisco, CA.

Felix Hill, KyungHyun Cho, Sébastien Jean, Coline Devin, and Yoshua Bengio. 2014. Not all neural embeddings are born equal. In *Proceedings of the NIPS Learning Semantics Workshop*, Montreal, Canada. Published online: `https://sites.google.com/site/learningsemantics2014/`.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of ACL Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126, Sofia, Bulgaria.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, pages 655–665, Baltimore, MD.

George Karypis. 2003. CLUTO: A clustering toolkit. Technical Report 02-017, University of Minnesota Department of Computer Science.

Sophia Katrenko and Pieter Adriaans. 2008. Qualia structures and their impact on the concrete noun categorization task. In *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics*, pages 17–24, Hamburg, Germany.

Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*, pages 423–430, Sapporo, Japan.

Alice Lai and Julia Hockenmaier. 2014. Illinois-lh: A denotational and distributional approach to semantics. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 329–334, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of ACL (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland.

Omer Levy and Yoav Goldberg. 2014b. Linguistic regularities in sparse and explicit word representations. In *Proceedings of CoNLL*, pages 171–180, Ann Arbor, MI.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. SemEval-2014 Task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of SemEval*, pages 1–8, Dublin, Ireland.

Oren Melamud, Ido Dagan, Jacob Goldberger, Idan Szpektor, and Deniz Yuret. 2014. Probabilistic modeling of joint-context in distributional similarity. In *Proceedings of CoNLL*, pages 181–190, Ann Arbor, MI.

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

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

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL*, pages 746–751, Atlanta, Georgia.

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

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Denis Paperno, Nghia The Pham, and Marco Baroni. 2014. A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of ACL*, pages 90–99, Baltimore, MD.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543, Doha, Qatar.

Klaus Rothenhäusler and Hinrich Schütze. 2009. Unsupervised classification with dependency based word spaces. In *Proceedings of the EACL GEMS Workshop*, pages 17–24, Athens, Greece.

Herbert Rubenstein and John Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Richard Socher, Brody Huval, Christopher Manning, and Andrew Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP*, pages 1201–1211, Jeju Island, Korea.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642, Seattle, WA.

Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, Cambridge, MA.

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

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Jiang Zhao, Tiantian Zhu, and Man Lan. 2014. Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 271–277, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.