Orthogonality of Syntax and Semantics within Distributional Spaces

Jeff Mitchell
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, UK
jeff.mitchell@ed.ac.uk

Mark Steedman
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, UK
steedman@inf.ed.ac.uk

Abstract

A recent distributional approach to word-analogy problems (Mikolov et al., 2013b) exploits interesting regularities in the structure of the space of representations. Investigating further, we find that performance on this task can be related to orthogonality within the space. Explicitly designing such structure into a neural network model results in representations that decompose into orthogonal semantic and syntactic subspaces. We demonstrate that learning from word-order and morphological structure within English Wikipedia text to enable this decomposition can produce substantial improvements on semantic-similarity, pos-induction and word-analogy tasks.

1 Introduction

Distributional methods have become widely used across computational linguistics. Recent applications include predicate clustering for question answering (Lewis and Steedman, 2013), bilingual embeddings for machine translation (Zou et al., 2013) and enhancing the coverage of POS tagging (Huang et al., 2013). The popularity of these methods, stemming from their conceptual simplicity and wide applicability, motivates a deeper analysis of the structure of the representations they produce.

Commonly, these representations are made in a single vector space with similarity being the main structure of interest. However, recent work by Mikolov et al. (2013b) on a word-analogy task suggests that such spaces may have further useful internal regularities. They found that semantic differences, such as between big and $small$, and also syntactic differences, as between big and $bigger$, were encoded consistently across their space. In particular, they solved the word-analogy problems by exploiting the fact that equivalent relations tended to correspond to parallel vector-differences.

In this paper, we investigate orthogonality between relations rather than parallelism. While parallelism serves to ensure that the same relation is encoded consistently, our hypothesis is that orthogonality serves to ensure that distinct relations are clearly differentiable. We focus specifically on semantic and syntactic relations as these are probably the most distinct classes of properties encoded in distributional spaces.

Empirically, we demonstrate that orthogonality predicts performance on the word-analogy task for three existing approaches to constructing word vectors. We also attempt to enhance the weakest of these three models by imposing an orthogonal structure in its construction. In these extensions, word representations decompose into orthogonal semantic and syntactic spaces, and we use word-order and morphology to drive this separation. This decomposition also allows us to define a novel approach to solving the word-analogy problems and our extended models become competitive with the other two original models. In addition, we show that the separate semantic and syntactic sub-spaces gain improved performance on semantic-similarity and POS-induction tasks respectively.

Our experiments here are based on models that construct vector-representations within a model that predicts the occurence of words in context. In particular we focus on the CBOW and Skip-gram models of Mikolov etal. (2013b) and Pennington et al.'s (2014) GloVe model. These models share the property of producing a single general representation for each word, which can be utilized in a variety of tasks, from POS tagging to semantic role labelling. In contrast, here we attempt to decompose the representations into separate seman-

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 1301–1310, Beijing, China, July 26-31, 2015.

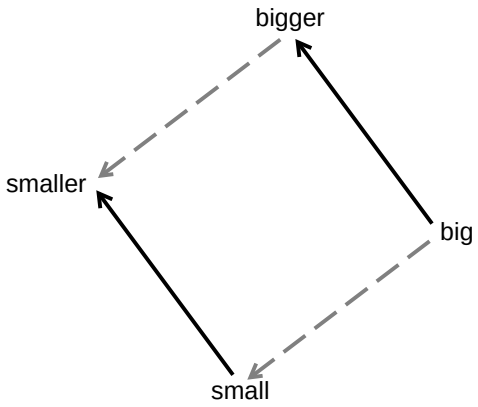

Figure 1: Geometric relationships between $small$, $smaller$, big and $bigger$.

tic and syntactic components.

To motivate this decomposition, consider the analogical reasoning task that Mikolov et al. (2013b) apply neural embeddings to. In this task, given vectors for the words big, $bigger$ and $small$, we try to predict the vector for $smaller$. They find that in practice $smaller \approx small + bigger - big$ produces an estimate that is frequently closer to the actual representation of $smaller$ than any other word vector. We can think of the vector $bigger - big$ as representing the syntactic relation that holds between an adjective and its comparative. Adding this syntactic structure to $small$ thus ends up at, or near, the relevant comparative, $smaller$. Alternatively, we could think of the vector $small - big$ as representing the semantic difference between small and big, and adding this relation to $bigger$ produces a semantic transformation to $smaller$.

Mikolov et al. (2013b) represent these sort of relations in terms of a diagram similar to Figure 1. The image places the four words in a 2D space and represents the relations between them in terms of arrows. The solid black arrows represent the syntactic relations $smaller - small$ and $bigger - big$, while the gray dashed arrows represent the semantic differences $smaller - bigger$ and $small - big$. Their solution to the analogy problem exploits the fact that these pairs of relations are approximately parallel to each other, i.e. that we can approximate $smaller - small$ with $bigger - big$, or $smaller - bigger$ with $small - big$. However, knowing that opposite sides of the square in Figure 1 are parallel to each other still leaves open the question of what happens at the corners. In other words, what is the relationship between the semantic differences, e.g. $smaller - bigger$, and the syntactic differences, e.g. $smaller - small$?

In this paper we explore the idea that such semantic and syntactic relations ought to be orthogonal to each other. This hypothesis arises both from the intuition that such distinct types of information ought to be represented distinctly within our space and also from the observation that solving the word-analogy task requires that words can be uniquely identified by combining these vector differences and so $small - big$ ought to be easily differentiable from $bigger - big$ as these relations point to different end results starting from big. Essentially, orthogonality will make better use of the volume within the space, spreading words with different semantic or syntactic characteristics further from each other.

In terms of predicting $smaller$ from big, $bigger$ and $small$, orthogonality of the relationship between $smaller - bigger$ and $smaller - small$ can be expressed in terms of their dot product:

$$(smaller - bigger) \cdot (smaller - small) = 0 \quad (1)$$

If all semantic relations were genuinely orthogonal to all syntactic relations, then their space would be decomposable into two orthogonal subspaces: one semantic, the other syntactic. Any word representation, $\mathbf{v}$, would then be the combination of a unique semantic vector, $\mathbf{b}$, within the semantic subspace and a unique syntactic vector, $\mathbf{s}$, within the syntactic subspace. If $\mathbf{b}$ were given a representation in terms of e components, and $\mathbf{s}$ in terms of f components, then $\mathbf{v}$ would have a representation in terms of $d = e + f$ components which would just be the concatenation of the two sets of components, which we will represent in terms of the operator $\oplus$.

$$\mathbf{v} = \mathbf{b} \oplus \mathbf{s} \quad (2)$$

Achieving this differentiation within the representations requires that the model have a means of differentiating semantic and syntactic information in the raw text. We consider two very simple approaches for this purpose, based on morphological and word order features. Both these types of features have been previously employed in simple word co-occurrence models (e.g., McDonald and Lowe, 1998; Clark, 2003), with bag-of-words and

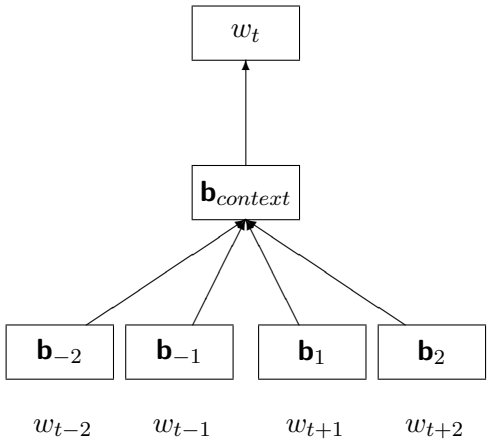

Figure 2: CBOW model predicting w_t from of a bag-of-words representation, $\mathbf{b}_{context}$, of a 4-word window around it.

lemmatization being good for semantic applications, while sequential order and suffixes is more useful for syntax. More recently, Mitchell (2013) demonstrated that word order could be used to separate syntactic from semantic structure, but only within a simple bigram language model, rather than a neural network model, and without exploiting morphology.

Our enhanced models are based on Mikolov et al.'s (2013a) CBOW architecture, which is described in Section 2. The novel extensions to it, employing a semantic-syntactic decomposition, are proposed in Section 3. We then describe our evaluation tasks and provide their results in Sections 5 and 6 respectively. These evaluations are based on the word-analogy dataset of Mikolov et al. (2013b), a noun-verb similarity task (Mitchell, 2013) and a POS clustering task.

2 Continuous Bag-of-Words Model (CBOW)

In the original CBOW model, the probability of a central target word, w_t, is predicted from a bag-of-words representation of the context it occurs in, as illustrated in Figure 2. This context representation, $\mathbf{b}_{context}$, is a simple sum of the CBOW vectors, $\mathbf{b}_i$, that represent each item, w_{t+i}, in a k-word window either side of the target.

$$\mathbf{b}_{context} = \sum_{i=-k, i\neq 0}^{k} \mathbf{b}_i \tag{3}$$

For speed, the output layer uses a hierarchical softmax function (Morin and Bengio, 2005). Each word is given a Huffman code corresponding to a path through a binary tree, and the output predicts the binary choices on nodes of the tree as independent variables. In comparison to the computational cost of doing the full softmax over the whole vocabulary, this hierarchical approach is much more efficient.

Each node is associated with a vector, $\mathbf{n}$, and the output at that node, given a context vector, $\mathbf{b}_{context}$, is:

$$p = logistic(\mathbf{n} \cdot \mathbf{b}_{context}) \tag{4}$$

Here, p is the probability of choosing 1 over 0 at this node of the tree, or equivalently finding a 1 in the Huffman code of w_t at the relevant position.

The objective function is the negative log-likelihood of the data given the model.

$$O = \sum -\log(p) \tag{5}$$

Where the sum is over tokens in the training corpus and the relevant nodes in the tree. Training is then based on stochastic gradient descent, with a decreasing learning rate.

3 Extensions

3.1 Continuous Sequence of Words (CSOW)

A major feature of the CBOW model is its use of a bag-of-words representation of the context and this is achieved by summing over the vectors representing words in the input. Although the model does seem to produce representations that are effective on both semantic and syntactic tasks, we want to be able to exploit word order information to separate these two characteristics. We therefore need to consider models which do not reduce the context to a structureless bag-of-words. Modifying the original model to retain the sequential information in the input is relatively straightforward. Instead of summing the input representations, we simply leave them as an ordered sequence of vectors, $\mathbf{s}_i$.

Then in the output layer, we require a vector for every input position, i, on every node. In this way, the output of the network depends on which context word is in which position, rather than just the set of words, irrespective of position in the input.

The network still learns a single representation for each word independently of position, but the output function has more parameters.

$$p = logistic(\sum_{i=-k,i\neq 0}^{k} \mathbf{n}_i \cdot \mathbf{s}_i) \quad (6)$$

Here each node of the tree is associated with one vector, $\mathbf{n}_i$, for each position, i, in the input context, giving $2k$ vectors in total at each node.

3.2 Continuous Bag and Sequence of Words (CBSOW)

Having introduced a sequential version of the CBOW model, what is really desired is a model that combines both bag and sequence components. Each word will have both an e-dimensional bag-vector $\mathbf{b}$ and an f-dimensional sequence-vector $\mathbf{s}$. The full representation of a word, $\mathbf{v}$, is then the concatenation of the components of $\mathbf{b}$ and $\mathbf{s}$.

Given this structure, the representation of a context of $2k$ words will be made up of the sum, $\mathbf{b}_{context}$, of their bag vectors, $\mathbf{b}_i$, as in the CBOW model given by Equation 3, along with the ordered sequence vectors, $\mathbf{s}_i$, as in the CSOW model. Each node in the tree then requires both a bag vector, $\mathbf{n}^b$, to handle the bag context, and $2k$ sequence vectors, $\mathbf{n}_i^s$, to handle the sequence context vectors, with probabilities given by:

$$p = logistic(\mathbf{n}^b \cdot \mathbf{b}_{context} + \sum_{i=-k,i\neq 0}^{k} \mathbf{n}_i^s \cdot \mathbf{s}_i) \quad (7)$$

3.3 Continuous Bag of Morphemes (CBOM)

A second source of information which might be used to differentiate semantic from syntactic representations is morphology. Specifically, English has the useful characteristic that the written words themselves can often be broken into a semantic stem on the left and a syntactic ending on the right. For example, *dancing* = *dance* + *ing* and *swimmer* = *swim* + *er*. In fact, stemming or lemmatization is commonly used in constructing distributional vectors precisely because throwing away the syntactic information helps to enhance their semantic content. Here, we want to use both the left and right halves separately to enhance both the semantic and syntactic components of the representations.

Our starting point is to break each word into a left-hand stem and a right-hand ending using CELEX (Baayen et al., 1995), as explained in more detail in Section 4.1.

The simplest model is then to represent each of these with its own vector, $\mathbf{l}_i$ and $\mathbf{r}_i$ respectively, and sum these vectors to form context representations of words in the input.

$$\mathbf{l}_{context} = \sum_{i=-k,i\neq 0}^{k} \mathbf{l}_i \quad (8)$$

$$\mathbf{r}_{context} = \sum_{i=-k,i\neq 0}^{k} \mathbf{r}_i \quad (9)$$

The output function takes much the same form as the original model but now each node needs both a left and a right vector, corresponding to the two context representations.

$$p = logistic(\mathbf{n}^l \cdot \mathbf{l}_{context} + \mathbf{n}^r \cdot \mathbf{r}_{context}) \quad (10)$$

3.4 Continuous Bag and Sequence of Words and Morphemes (CBSOWM)

Finally, we want to incorporate all these elements in a single model, with the morphological and word order elements of the model working in harmony. In particular, we want the sequential part of the model to be guided by morphological information without being constrained to give all words with same ending the same representation. Our solution is to add a constraint term to the objective function, which penalizes sequence vectors that stray far from the relevant morphological representation. The bag vectors, in contrast, are determined directly by the left hand stems, with all words having the same stem then sharing the same bag vector, $\mathbf{b} = \mathbf{l}$.

The main structure of the model remains as in the CBSOW model, with the context being represented by the sum of bag vectors alongside the ordered sequence vectors. Output probabilities are as given by Equation 7, and we add a morphological penalty, m, to the objective function.

$$m = \sum_{i=-k,i\neq 0}^{k} \frac{1}{2}\lambda|\mathbf{s}_i - \mathbf{r}_i|^2 \quad (11)$$

The morphological representations $\mathbf{r}$ enter into the model only through the penalty term, and they adapt during training solely in terms of this interaction with the sequence vectors. Gradient descent results in the $\mathbf{r}$ vectors moving towards the centre of the corresponding $\mathbf{s}$ vectors, and the $\mathbf{s}$ vectors in turn being drawn towards that centre.

The result is to elastically connect all the **s** vectors corresponding to a single morphological element through their **r** vectors, so that they are drawn together, but can still develop idiosyncratically if there is sufficient evidence in the data.

3.5 Application to the Word-Analogy Task

Decomposition of representations into separate semantic and syntactic spaces enables us to utilise a new approach to solving the word-analogy problems. Rather than using vector differences to predict a vector, we can instead construct it by copying the relevant bag and sequence vectors. So, since $small$ and $smaller$ share very similar semantic content, we can use the bag vector of $small$ as the bag vector of $smaller$, since that is where the semantic content is mainly represented: $\mathbf{b}_{smaller} \approx \mathbf{b}_{small}$. Similarly, we can use the sequence vector of $bigger$ as the sequence vector for $smaller$, since these words share common syntactic behaviour: $\mathbf{s}_{smaller} \approx \mathbf{s}_{bigger}$.

The predicted representation of $smaller$ is then given by the concatenation of the components.

$$\mathbf{v}_{smaller} \approx \mathbf{b}_{small} \oplus \mathbf{s}_{bigger} \tag{12}$$

We find that this gives the best performance on the models that use word-order features (CBSOW and CBSOWM).

4 Training

Our experiments are based on the publicly available word2vec[1] and GloVe[2] packages. We modified the original CBOW code to incorporate the CBSOW, CBOM and CBSOWM extensions described above, and trained models on three English Wikipedia corpora of varying sizes, including the enwik8 and enwik9 files[3] suggested in the word2vec documentation, containing the first 10^8 and 10^9 characters of a 2006 download, and also a full download from 2009. On the smallest 17M word corpus we explored a range of vector dimensionalities from 10 to 1000. On the larger 120M and 1.6B word corpus, we trained extended models with a 200-dimensional semantic component and a 100-dimensional syntactic component comparing to 300-dimensional CBOW, Skip-gram and GloVe models. The parameter, λ, in Equation 11 was set to 0.1 and the recommended window sizes of 5, 10 and 15 words either side of the central word were used as context for the CBOW, Skip-gram and GloVe models respectively.

[1] https://code.google.com/p/word2vec/

[2] http://nlp.stanford.edu/projects/glove/

[3] http://mattmahoney.net/dc/text.html

4.1 CELEX

We attempted to split all the words in the training data into a left hand and a right hand using CELEX (Baayen et al., 1995), an electronic dictionary containing morphological structure. In the cases of words that were not found in the dictionary and also those that were found but had no morphological substructure, the left hand was just the whole word and the right hand was a $-NULL-$ token. For the remaining words, we treated short suffixes as being syntactic inflections and stripped all these off to leave a left hand 'semantic' component. The 'syntactic' component was then rightmost of these suffixes, with any additional suffixes being ignored.

5 Evaluation

The hypothesis that orthogonality is useful to word vector representations is investigated empirically in two ways. Firstly, we attempt to quantify the orthogonality that is already implicitly present in the original CBOW, Skip-gram and GloVe representations and relate that to their success in the word-analogy task. Secondly, the extensions described above are evaluated on a number of tasks in order to evaluate the benefits of their explicit orthogonality between components.

5.1 Orthogonality within the Original Models

Equation 1 relates orthogonality of vector differences to their dot product being zero, which corresponds to the fact the cosine of 90° is zero. Thus, we can use the cosine as a quantification of how close to orthogonal the vector differences are and then relate that to performance on the word-analogy dataset distributed with the word2vec toolkit.

That task involves predicting a word vector given vectors for other related words. So, for example, given vectors for big, $bigger$ and $small$, we would try to predict a vector for $smaller$. We then judge the success of this prediction in terms of whether the predicted vector is in fact closer to $smaller$'s actual word vector than any other word vector. The dataset contains 19,544 items, broken down into 14 subtasks (e.g. capitals of common countries or adjective to adverb conversion).

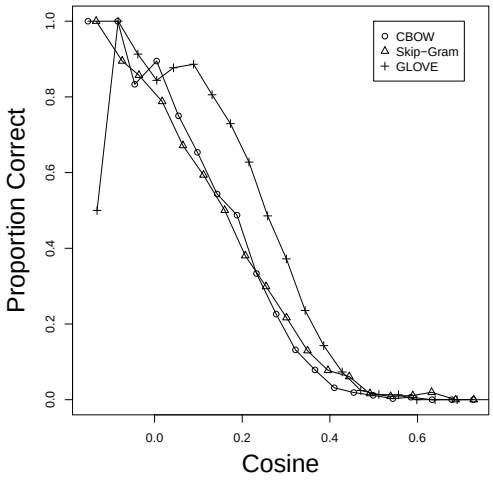

Figure 3: Proportion Correct against Average Cosine.

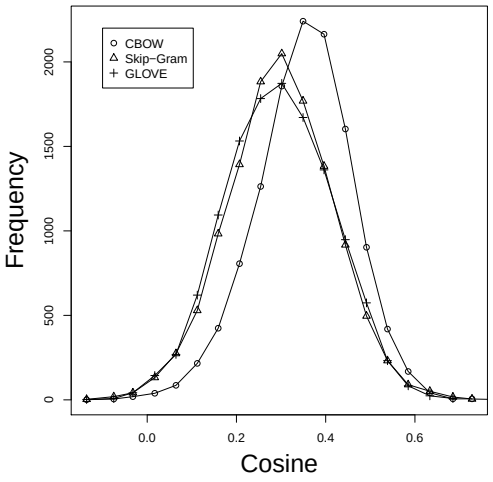

Figure 4: Frequency against Cosine.

For each item, we measure the cosine of the angle between the vector differences for the word we are trying to predict (e.g. $smaller - small$ and $smaller - bigger$) and analyze these values in terms of the success of the model's prediction, with smaller cosine values corresponding to angles that are closer to orthogonal.

5.2 CBOW Extensions

We evaluate the extensions on three tasks. Alongside the word-analogy problems, we also evaluate the separate semantic and syntactic sub-spaces on their own individual tasks. The semantic task correlates predicted semantic similarities with the noun-verb similarity ratings gathered by Mitchell (2013), and the remaining task clusters the syntactic representations and evaluates these clusters in relation to the POS classes found in the Penn Treebank.

On the word-analogy problem we compare to the original CBOW, Skip-gram and GloVe models. In the case of these original models and also the CBOM model, we follow Mikolov et al.'s (2013b) method for making the word-analogy predictions in terms of addition and subtraction: $smaller \approx bigger - big + small$. However, in the case of the CBSOW and CBSOWM models, we use the novel approach described in Section 3.5: $\mathbf{v}_{smaller} \approx \mathbf{b}_{small} \oplus \mathbf{s}_{bigger}$. Similarity is then based on the cosine measure for all types of representation.

The noun-verb similarity task is based on correlating the model's predicted semantic similarity for words with human ratings gathered in an online experiment. Such evaluations have been commonly used to evaluate distributional representations, with higher correlations indicating a model which is more effective at forming vectors whose relations to each other mirror human notions of semantic similarity. Mitchell (2013) argued that predicting semantic similarity relations across syntactic categories provided a measure of the extent to which word representations succeed in separating semantic from syntactic content, and gathered a dataset of similarities for noun-verb pairs. Each rated item consists of a noun paired with a verb, and the pairs are constucted to range from high semantic similarity, e.g. *disappearance - vanish*, to low, e.g. *transmitter - grieve*. The dataset contains ratings for 108 different pairs, each of which was rated by 20 participants. For the CBOW model, we predict similarities in terms of the cosine measure for the two word vectors. For the other models, we predict similarities from cosine applied to just the bag or left-hand vectors.

The syntactic component of the representations is evaluated by clustering the vectors and then comparing the induced classes to the POS classes found in the Penn Treebank. We use the many-to-one measure (Christodoulopoulos et al., 2010; Yatbaz et al., 2012) to determine the extent to which the clusters agree with the POS classes. Each cluster is mapped to its most frequent gold tag and the reported score is the proportion of word tokens correctly tagged using this mapping. The clustering itself is a form of k-means clustering, where similarity is measured in terms of the cosine measure. Each vector is assigned to a clus-

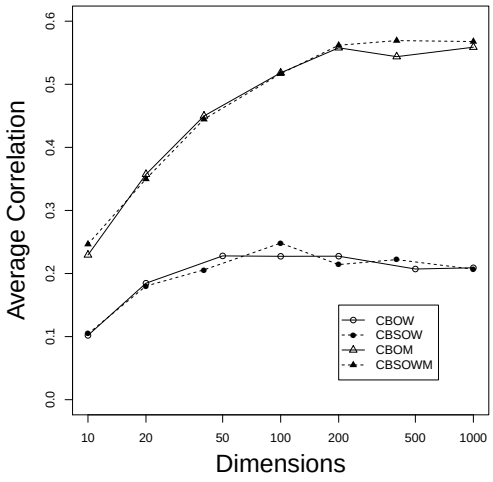

Figure 5: Average Correlation on Noun-Verb Evaluation Task against Size of Representations.

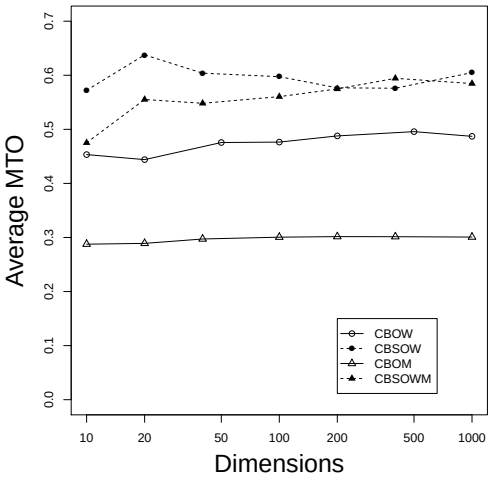

Figure 6: Average Many-To-One Evaluation against Size of Representations.

ter based on which cluster centroid it is most similar to and then the cluster centroids are updated given the new cluster assignments and the process repeats. This clustering was applied to either the sequence or right-hand vectors in the case of the CBSOW, CBOM and CBSOWM models, and to the whole vectors in the case of CBOW. We randomly initialized 45 clusters and then evaluated after 100 iterations of the k-means algorithm.

6 Results

6.1 Original Models

Figure 3 is a plot of the proportion of correct predictions made by 100-dimensional CBOW, Skip-Gram and GloVe models on the word-analogy task against cosine of the angle between the vector differences. The range of the cosine distribution was broken into twenty intervals and the plotted values were derived by calculating the proportion correct and average cosine value within each interval. It is clear from the resulting curves that cosine is a fairly strong predictor for all models of whether the model gets a word-analogy item correct, with higher rates of success for smaller cosine values - i.e. angles closer to orthogonality. This is confirmed by a significant ($p < 0.001$) result from a logistic regression of correctness against cosine value. Similar results are found for both the semantic subtasks (e.g. capitals of common countries) and syntactic subtasks (e.g. adjective to adverb conversion) considered separately.

The actual distribution of cosine values for each type of model is given in Figure 4. This analysis reveals that while the Skip-Gram and GloVe models have fairly similar cosine distributions, the CBOW model's distribution is shifted to the right, with more angles further from othogonality. This begs the question of what the effect on performance would be if we managed to push more of the CBOW distribution towards zero, and in the next section we examine the extensions that implement this idea.

6.2 CBOW Extensions

We first consider the models trained on the smaller 17M word corpus, and the evaluations of these models on the noun-verb similarity and POS clustering tasks are presented in Figures 5 and 6 respectively. These graphs depict the performance as the representations grow in size. For the CBOW model, this is just the dimension of the induced vectors. For the other models, we consider models with equal sizes of semantic and syntactic subspaces and report performance against the total dimensionality of the combined representation. For both these tasks, the results were averaged over ten repetitions of training with random initializations.

On the noun-verb similarity task, morphology produces the largest performance gains, with the CBOM model substantially outperforming the CBOW model. Word order structure has no clear impact.

On the syntactic task, in contrast, it is word order that produces reliable gains, with the CBSOW model clearly improving on the CBOW model. The simplistic use of morphology in the CBOM model results in a degradation of performance in

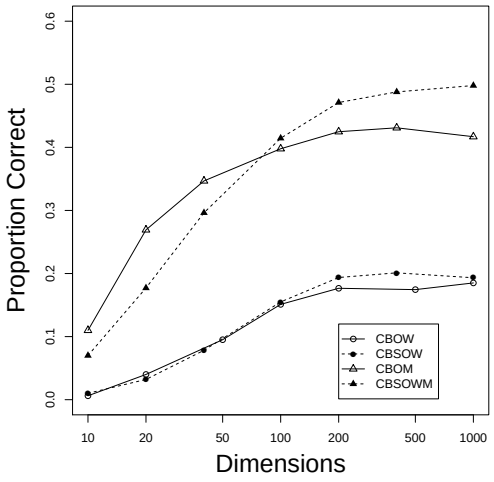

Figure 7: Proportion Correct on the Analogy Task against Size of Representations.

comparison to the CBOW model, but the CBSOWM model's performance is comparable to that of the CBSOW for larger representations.

Thus for these two tasks, the CBSOWM results appear to show a reasonable integration of morphology and word order information giving good performance on both semantic and syntactic tasks. This conclusion is borne out the results of the word-analogy tasks in Figure 7, where the CBSOWM model outperforms all the other models. Here, morphology gives the greatest benefit on its own, as evidenced in the differences between the CBOW and CBOM models. Nonetheless, word order still produces noticeable improvements, with the CBSOW result beating the CBOW results, and the CBSOWM beating the CBOM at larger dimensions. There is considerable variation in the effects on performance among the various analogy subtasks, but even a task such as capitals of common countries (e.g. predicting Iraq as having Baghdad as its capital, given that Greece has Athens) appears to benefit from decomposition of representations, despite not obviously involving syntactic structure.

Table 1 compares 300-dimensional models across different sizes of training data. In the case of the CBSOW, CBOM and CBSOWM models we use representations with 200 semantic and 100 syntactic dimensions and compare these to CBOW, Skip-gram and GloVe models of the same total size. It is clear for all quantities of training data that all the extensions outperform the basic CBOW model, with morphology giving greater

	Training Words		
Model	17M	120M	1.6B
GloVe	29.53%	58.18%	**72.54%**
Skip-Gram	30.03%	52.67%	62.34%
CBOW	18.47%	38.48%	54.17%
CBSOW	20.83%	42.00%	59.41%
CBOM	44.29%	53.60%	61.87%
CBSOWM	**48.92%**	**63.19%**	68.32%

Table 1: Performance of 300-Dimensional Models on the Word-Analogy Task

gains than word order, and the combined CBSOWM model outperforming both. This performance advantage of the CBOM over CBSOW appears to weaken as the training data grows, which is probably the effect of both the lack of morphological information for rare words encountered in the larger datasets and also the diminishing returns on that information as more data provides better supervision of the training process. The sequential information, in contrast, is internal to the training data and seems to provide the same, or greater, performance boost as the training set grows.

Comparing the results of our extended models to the Skip-gram and GloVe models, we can see that on the two smaller corpora CBSOWM outperforms both these models, while on the largest corpus, it only beats the Skip-gram results and GloVe achieves the best performance. Of course, neither the Skip-gram nor GloVe models has access to the morphological information that the CBSOWM model uses, but the results demonstrate that the performance of the CBOW model can be substantially boosted by exploiting a representational structure that decomposes into semantic and syntactic sub-spaces. Similar methods could in principle be applied to most word embedding models, including Skip-gram and GloVe.

We can also examine the distribution of cosine values for the new models. Figure 8 compares the distribution of cosine values for CBOW, CBSOW, CBOM and CBSOWM models. Although, in comparison to the original CBOW model, each of the extended models shifts the distribution towards zero, i.e. towards orthogonality, this shift for the CBSOW model is marginal. In contrast, the CBOM model has a large number of instances where the cosine is exactly zero, corresponding to cases where all of the relevant morphological information is found in CELEX. The remainder

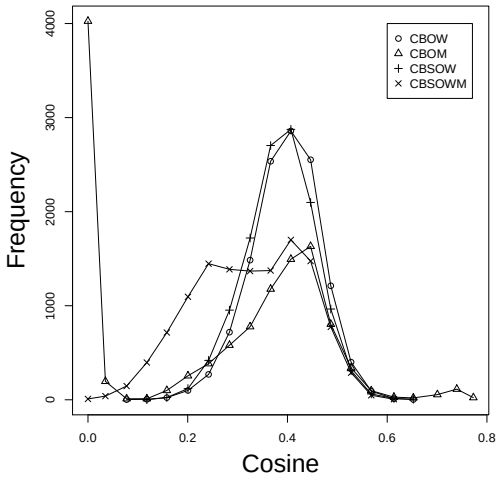

Figure 8: Frequency against Cosine.

of the data, however, seems to be less orthogonal than the original CBOW distribution, suggesting that words without a morphological analysis need a more sophisticated treatment. The shift in the CBSOWM distribution, in comparison, is less radically bimodal, with more continuity between those words with and without morphology. This reflects the difference in these models handling of suffixes, with the CBSOWM model's greater flexibility resulting in gains over the CBOM model on the POS induction and word analogy tasks.

7 Conclusions

Our experiments demonstrate the utility of orthogonality within vector-space representations in a number of ways. In terms of existing models, we find that the cosines of vector-differences is a strong predictor of the performance of CBOW, Skip-gram and GloVe representations on the word analogy task, with smaller cosine values - corresponding to angles closer to orthogonality - being associated with a greater proportion of correct predictions. With regard to developing new models, this orthogonality of relationships inspired three models which used word-order and morphology to separate semantic and syntactic representations. These separate sub-spaces were shown to have enhanced performance in semantic similarity and POS-induction tasks and the combined representations showed enhanced performance on the word-analogy task, using a novel approach to solving this problem that exploits the decomposable structure of the representations.

Both Botha and Blunsom (2014) and Luong et al. (2013) take a more sophisticated approach to morphology[4], constructing a word's embedding by recursively combining representations of all its morphemes, though only within a single non-decomposed space. Future work ought to pursue models in which all morphemes contribute both semantic and syntactic content to the word representations.

It would also be desirable to explore more practical applications of these representations than the limited evaluations presented here. It seems feasible that our decomposition of representations could benefit tasks that need to differentiate their treatment of semantic and syntactic content. In particular, applications of word embeddings that mainly involve syntax, such as POS tagging (e.g., Tsuboi, 2014) or supertagging for parsing (e.g., Lewis and Steedman, 2014), may be a reasonable starting point.

Acknowledgements

We would like to thank Stella Frank, Sharon Goldwater and other colleagues along with our reviewers for criticism, advice and discussion. This work was supported by ERC Advanced Fellowship 249520 GRAMPLUS and EU Cognitive Systems project FP7-ICT-270273 Xperience.

References

Harald Baayen, Richard Piepenbrock, and Hedderik van Rijn. 1995. CELEX2 LDC96L14. Web Download. Philadelphia: Linguistic Data Consortium.

Jan A. Botha and Phil Blunsom. 2014. Compositional Morphology for Word Representations and Language Modelling. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, Beijing, China.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised POS induction: How far have we come? In *Proceedings of EMNLP*, pages 575–584.

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the tenth Annual Meeting of the European Association for Computational Linguistics (EACL)*, pages 59–66.

[4]Though not neccessarily better performing. Luong et al.'s published 50-dimensional embeddings trained on 986M words scored only 13.57% on the word-analogy task, well behind 40-dimensional CBOM (34.68%) and CBSOWM (36.71%) models trained on 17M words.

Fei Huang, Arun Ahuja, Doug Downey, Yi Yang, Yuhong Guo, and Alexander Yates. 2013. Learning Representations for Weakly Supervised Natural Language Processing Tasks. *Computational Linguistics*, 40:85–120.

Mike Lewis and Mark Steedman. 2013. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000, Doha, Qatar, October. Association for Computational Linguistics.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria, August. Association for Computational Linguistics.

Scott McDonald and Will Lowe. 1998. Modelling functional priming and the associative boost. In *Proceedings of the 20th Annual Meeting of the Cognitive Science Society*, pages 675–680. Erlbaum.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

Jeff Mitchell. 2013. Learning semantic representations in a bigram language model. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Short Papers*, pages 362–368, Potsdam, Germany, March. Association for Computational Linguistics.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *AISTATS05*, pages 246–252.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Yuta Tsuboi. 2014. Neural networks leverage corpus-wide information for part-of-speech tagging. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 938–950, Doha, Qatar, October. Association for Computational Linguistics.

Mehmet Ali Yatbaz, Enis Sert, and Deniz Yuret. 2012. Learning syntactic categories using paradigmatic representations of word context. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 940–951, Jeju Island, Korea, July. Association for Computational Linguistics.

Will Y. Zou, Richard Socher, Daniel M. Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398. ACL.