Assessing Composition in Sentence Vector Representations

Allyson Ettinger[1], **Ahmed Elgohary**[2], **Colin Phillips**[1], **Philip Resnik**[1,3]
[1]Linguistics, [2]Computer Science, [3]Institute for Advanced Computer Studies
University of Maryland, College Park, MD
{aetting,colin,resnik}@umd.edu, elgohary@cs.umd.edu

Abstract

An important component of achieving language understanding is mastering the composition of sentence meaning, but an immediate challenge to solving this problem is the opacity of sentence vector representations produced by current neural sentence composition models. We present a method to address this challenge, developing tasks that directly target compositional meaning information in sentence vector representations with a high degree of precision and control. To enable the creation of these controlled tasks, we introduce a specialized sentence generation system that produces large, annotated sentence sets meeting specified syntactic, semantic and lexical constraints. We describe the details of the method and generation system, and then present results of experiments applying our method to probe for compositional information in embeddings from a number of existing sentence composition models. We find that the method is able to extract useful information about the differing capacities of these models, and we discuss the implications of our results with respect to these systems' capturing of sentence information. We make available for public use the datasets used for these experiments, as well as the generation system.[1]

1 Introduction

As natural language processing strives toward language understanding, it is important that we develop models able to extract and represent the *meaning* of sentences. Such representations promise to be applicable across a variety of tasks, and to be more robust than non-meaning-based representations for any given task requiring meaning understanding. To accomplish meaning extraction, a particular need is that of mastering composition: systematic derivation of the meaning of a sentence based on its parts.

In this paper we tackle compositional meaning extraction by first addressing the challenge of evaluation and interpretability: after all, in order to improve meaning extraction, we need to be able to evaluate it. But with sentence representations increasingly taking the form of dense vectors (embeddings) from neural network models, it is difficult to assess what information these representations are capturing—and this problem is particularly acute for assessing abstract content like compositional meaning information.

Here we introduce an analysis method for targeting and evaluating compositional meaning information in sentence embeddings. The approach builds on a proposal outlined in Ettinger et al. (2016), and involves designing classification tasks that directly target the information of interest (e.g., "Given a noun n, verb v, and an embedding $\mathbf{s}$ of sentence s: is n the *agent* of v in s?"). By contrast to related work analyzing surface variables like word content and word order in sentence embeddings (Adi et al., 2016), we specifically target compositional meaning information relevant to achieving language understanding—and in order to isolate this more abstract information, we exert careful control over our classification datasets to ensure that we are targeting information arrived at by composition of the source

[1]Code for the generation system, as well as a pointer to the classification datasets, can be found at https://github.com/aetting/compeval-generation-system

Proceedings of the 27th International Conference on Computational Linguistics, pages 1790–1801
Santa Fe, New Mexico, USA, August 20-26, 2018.

sentence, rather than general statistical regularities. Our approach is informed by methods in cognitive neuroscience and psycholinguistics, where such controls are standard practice for studying the brain.

In particular, to ensure validity of our tests we introduce three mechanisms of control. First, to create controlled datasets at the necessary scale, we develop a generation system that allows us to produce large sentence sets meeting specified semantic, syntactic and lexical constraints, with gold-standard meaning annotation for each sentence. Second, we control the train-test split so as to require more robust generalization in order to perform the tasks successfully. Third, we employ a sanity check leveraging known limitations of bag-of-words (BOW) composition models: for any tasks requiring order information *from the source sentence*, which BOW models cannot logically retain, we check to ensure that BOW composition models are at chance performance.

These controls serve to combat a problem that has gained increasing attention in recent work: many existing evaluation datasets contain biases that allow for high performance based on superficial cues, thus inflating the perceived success of systems on these downstream tasks (Gururangan et al., 2018; Bentivogli et al., 2016). In the present work, our first priority is careful control of our tasks such that biases are eliminated to the greatest extent possible, allowing more confident conclusions about systems' compositional capacities than are possible with existing metrics.

The contributions of this paper are threefold. 1) We introduce a method for analyzing compositional meaning information in sentence embeddings, along with a generation system that enables controlled creation of datasets for this analysis. 2) We provide experiments with a range of sentence composition models, to demonstrate the capacity of our method to shed light on compositional information captured by these models. 3) We make available the classification datasets used for these experiments, as well as the generation system used to produce the sentence sets, to allow for broader testing of composition models and to facilitate creation of new tasks and classification datasets.

Although we focus on neural composition models and sentence embeddings in the present paper—due to the current dominance of these methods and the need to evaluate their compositional capacities—it is important to note that this analysis method can also be applied more broadly. Since the method simply operates by classification of sentence representations, it can be applied to any format of sentence representation that can be input as features to a classifier.

2 Meaning and composition

In this section we will briefly explain the concepts of *meaning* and *composition*, which are the central targets of our analyses in this work.

Our approach assumes there to be identifiable components of *meaning* that we can expect in well-formed sentence representations. For instance, the sentence "the dog chased the girl" contains the information that there was a chasing event, and a dog was the chaser (*agent* of chasing) and a girl the chasee (*patient* of chasing). The sentence "the dog did not bark" conveys that a barking event did not happen.

Humans are able to extract meaning with remarkable robustness, and a key factor in human language understanding is *composition*: the productive combinatory capacity that allows sentence meaning to be derived systematically based on the meanings of its parts (Heim and Kratzer, 1998). To illustrate the power of this systematicity, consider a nonsensical sentence like the following:

The turquoise giraffe recited the sonnet but did not forgive the flight attendant.

Though this sentence describes an entirely implausible scenario, and though nothing like it should ever have occurred in any corpus or conversation, any English speaker is able to extract the meaning of this sentence without difficulty. This is because language is highly systematic, and the meanings of the parts of the sentence can be combined predictably to arrive at the full meaning.

Regardless of how closely NLP systems should draw on human strategies for language processing, the need for composition is clear: if systems do not construct meanings of sentences based on their parts, then the alternative is memorization of all possible sentences, which is neither practical nor possible.

In this work, critically, we are focused on the results of systematic compositional processes, to be distinguished from biases based on general statistical regularities. The importance of this distinction is

highlighted by the result reported in Adi et al. (2016), which shows a BOW composition model attaining 70% accuracy on a binary word order classification task. This result is surprising given that BOW models (which simply average together word-level representations) necessarily sacrifice any order information from the source sentence. This suggests that the above-chance performance relies on statistical regularities of word ordering in the data as a whole, independent of the source sentence—that is, the model's above-chance performance must be dependent on some correspondence between word orders being tested and word orders seen when training the word embeddings.

Although sensitivity to such regularities is often useful, in this work we are concerned with *systematic composition of the source sentence itself*, abstracting away from general statistical regularities. This is critical for our purposes: to master composition, models must be able to construct the meaning of a sentence not only when it matches commonly-seen patterns (e.g., "the cat chased the mouse") but also when it deviates from such patterns (e.g., "the mouse chased the cat"). This is the reasoning behind our BOW sanity check, discussed in Section 3.3, which serves to ensure that our tests cannot be solved by simple averaging. Additionally, the biases in naturally-occurring data, further highlighted by the Adi et al. result, motivate our use of generated data for the sake of maintaining the necessary level of control.

3 The present method

3.1 Approach

The approach that we take to probe for compositional meaning information in sentence embeddings is inspired by the neuroscience technique of multivariate pattern analysis (Haxby et al., 2014), which tests for encoding of information in patterns of neural data by means of classification tasks designed to be contingent on the information of interest. Our use of careful control in implementing this approach is also informed more generally by the methodologies of cognitive neuroscience and psycholinguistics, which standardly use these kinds of controls to draw conclusions about information in human brain activity. The approach that we develop here builds on the proposal of Ettinger et al. (2016)—which described the basic form of the method and provided a simple validation with a small set of active and passive sentences. In the present work we flesh out and strengthen the method with a number of more rigorous controls aimed at better isolating the information of interest, and we substantially expand the scope of the tests through the use of a more sophisticated sentence generation system.

3.2 Classification tasks

As proposed by Ettinger et al. (2016), we target two meaning components as our starting point: semantic role and negation. These components are priorities because they are fundamental to the meaning of a sentence, having bearing on the key questions of "what happened (and what didn't)" and "who did what to whom". Additionally, they represent information types that can be heavily distorted with respect to surface variables like word content and order: to know semantic role and negation information, it is not enough to know which words are in the sentence or which words come earlier in the sentence.

We formulate the semantic role classification task ("**SemRole**") as follows: "Given representation $\mathbf{n}$ of probe noun n, representation $\mathbf{v}$ of probe verb v, and embedding $\mathbf{s}$ of sentence s (with s containing both n and v), does n stand in the AGENT relation to v in s?" For example, an input of $\{n$: "professor", v: "help", s: "the professor helped the student"$\}$ would receive a positive label because *professor* is AGENT of *help* in the given sentence.

We formulate the negation classification task ("**Negation**") as follows: "Given a representation $\mathbf{v}$ of a probe verb v, and an embedding $\mathbf{s}$ of sentence s (with s containing v, one negation, and one other verb), is v positive or negated in s?" For example, an input of $\{v$: "sleep", s: "the professor is not actually helping the student who is totally sleeping"$\}$ receives a positive label because *sleep* is not negated in that sentence. To decouple this from a simpler task of identifying adjacency between negation and a verb, we insert variable-length adverb sequences (e.g., *not **really, actually** helping*) before the verbs in the dataset (negated and non-negated), to ensure that the negation is not always adjacent to the verb that it affects.

These formulations differ from those in the original Ettinger et al. (2016) proposal, instead making use of variable word probes as employed by Adi et al. (2016). This adjustment was made to maximize

the generalization required for strong performance on the tasks, and to further reduce vulnerability to biasing cues in the datasets. More detail on our implementation of this formulation is given in Section 5.

3.3 Means of control

The most critical consideration in this work is ensuring that we can draw valid conclusions about composition from performance on our classification tasks. To this end, we take a number of measures to control our data, to avoid biasing cues that would make the tasks solvable independent of the information of interest—a problem observed in many existing datasets, as mentioned above (Gururangan et al., 2018).

Generation system A critical component of isolating abstract meaning information is employing syntactic variation, such that the meaning information of interest is the single underlying variable distinguishing label categories. For instance, we might use sentences like "the professor helped the student", "the student was helped by the professor", and "the student that the professor helped was sleeping"—which vary in structure, but which share an underlying event of a professor helping a student.

In order to produce sentence sets that exhibit this level of variation—and that reach the necessary scale for training and testing classifiers—without allowing the statistical biases of naturally-occurring data, we developed a generation system that takes as input lexical, semantic and syntactic constraints, and that produces large sentence sets meeting those constraints. In addition to allowing us to produce controlled datasets, this system also ensures that the generated datasets are annotated with detailed semantic and syntactic information. This generation system is described in greater detail in Section 4.

Train/test splits To be confident that the classifier is picking up on underlying meaning information and not simply a union of different superficial cues across syntactic structures, we make careful provisions in our train/test split to ensure generalization (beyond the obvious split such that sentences in test do not appear in training). For our semantic role task, certain (n,v) probe combinations are held out for test, such that no combinations seen at test time have been seen during training. This is done to ensure that the classifier cannot rely on memorized sequences of words. For our negation task, which uses only one probe, we hold out certain adverbs from training (as described above, adverbs are used as material to separate the negation and the verb), such that at test time, the material separating the negation and the verb (or preceding the non-negated verb) has never been seen in training.

BOW as control As described above, it is logically impossible for BOW models to encode information that requires access to word order from the source sentence itself. We leverage this knowledge to create a sanity check baseline for use in monitoring for lexical biases: if, for any task requiring access to word order information, the BOW baseline performs above chance, we know that the datasets contain lexical biases affecting the classification results, and we can modify them accordingly.

4 Generation system

In this section we describe the generation system that we use to create large, controlled datasets for our classification tasks. As described above, this system takes input constraints targeting semantic, syntactic, and lexical components, and produces diverse, meaning-annotated sentences meeting those constraints.

4.1 Event/sentence representations

As a framework for specifying semantic and syntactic constraints, we use a class of event representations that contain both lexicalized semantic information and necessary syntactic information, such that there is a deterministic mapping from a fully-populated event representation to a corresponding surface sentence form. These representations fall roughly within the category of "lexicalized case frame" outlined by Reiter and Dale (2000) for natural language generation. Figure 1 shows an example representation, in fully-specified textual form, and in simplified graphical form.

Our representations are currently restricted to events denoted by transitive and intransitive verbs, with the arguments of those verbs and optional transitive or intransitive relative clauses on those arguments.

These representations are comparable in many ways to abstract meaning representation (AMR) (Banarescu et al., 2012), but rather than abstracting entirely away from syntactic structure as in AMR, our

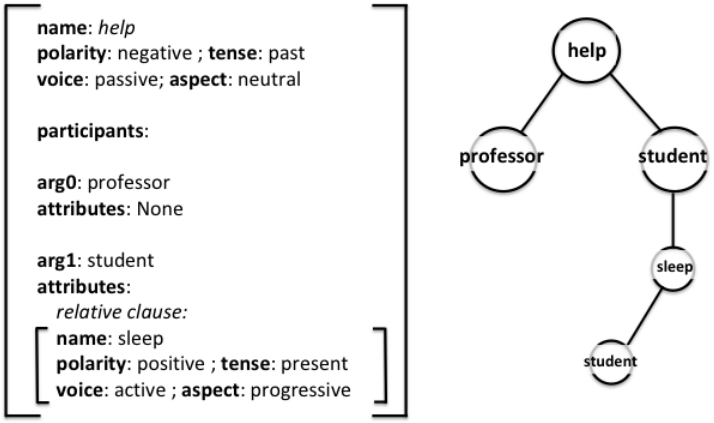

Figure 1: Event representation for *"The student who is sleeping was not helped by the professor"*

event representations encode syntactic information directly, along with the more abstract meaning information, in order to maintain a deterministic mapping to surface forms. Relatedly, while AMR uses PropBank frames (Palmer et al., 2005) to encode meaning information, we encode information via English lemmas, to maintain control over lexical selection during generation.

These representations can be partially specified to reflect a desired constraint, and can then be passed in this partial form as input to the generation system—either as a required component, or as a prohibited component. This allows us to constrain the semantic and syntactic characteristics of the output sentences. In addition to partial events, the system can also take lists of required or prohibited lexical items.

4.2 Event population

The system uses a number of structural templates into which partial events can be inserted. Structural templates vary based on the transitivity of verbs and the presence or absence of relative clauses on arguments—for instance, if the nodes in the right side of Figure 1 were unpopulated, it would depict an empty structural template consisting of a transitive main verb with an intransitive relative clause on arg1. Once we have inserted a partial event into a subsection of an empty structural template (events can be inserted into either the main clause or a relative clause), the system populates the remainder of the event components by iterating through available verbs and nouns of the vocabulary, and through available values for unfilled syntactic characteristics (such as polarity, tense, voice, etc.).

For simplicity, we control plausibility of argument/predicate combinations by setting the system vocabulary such that it contains only animate human nouns, and only verbs that can take any of those nouns in the relevant argument slots. This is a reasonable task due to the capacity of the system to generate thousands of sentences from only a handful of nouns and verbs. We leave incorporation of more sophisticated selectional preference methods (Van de Cruys, 2014; Resnik, 1996) for future work.

Our goal is to find the optimal balance between the critical need of this method for structurally variable, carefully controlled sentences, and the practical need to avoid substantial deviation from sentence types to which systems will have been exposed during training. To this end, we draw our vocabulary from comparatively frequent words, and we impose structural constraints to limit the complexity of sentences—specifically, in the current experiments we restrict to sentences with no more than one relative clause, by omitting templates that include relative clauses on both arguments of a main verb.

4.3 Syntactic realization

Once an event representation is fully populated, it is submitted to a surface realization module that maps from the event to a surface sentence via a simple rule-based mapping. Since the representations specify syntactic information and use lexicalized meaning information, there is no significant process of lexical selection required during surface realization—only morphological inflection derivable from syntactic characteristics. As a result, the event representations map deterministically to their corresponding surface

forms. We use a grammar specified using the NLTK feature grammar framework (Bird et al., 2009). Morphological inflections are drawn from the XTAG morphological database (Doran et al., 1994).

4.4 Sentence quality

To ensure the quality of generation system output, we manually inspected large samples of generated sentences throughout development and after generation of the final sets, to confirm that sentences were grammatical and of the expected form. Table 1 shows a representative sample of generated sentences.[2]

the men were sleeping
the woman followed the lawyer that the student is meeting
the women were being helped by the lawyers
the student called the man
the scientist that the professors met is dancing
the doctors that helped the lawyers are being recommended by the student

Table 1: Example generated sentences

5 Implementation of lexical variability

As discussed above, we adopt the variable probe formulation used by Adi et al. (2016). This adds a dimension to the learning task that is not present in the original Ettinger et al. (2016) task formulation: the classifier needs not only to identify meaning information in the input sentence—it needs to identify meaning information contingent on the identities of the particular probe words.

To identify the probe word(s) in the input features, Adi et al. (2016) use the source word embeddings, but this is problematic for our purposes, given that we want to test a wide variety of models, which use word embeddings of different types and sizes. To avoid this variability, it would be preferable to use one-hot vectors to identify word probes. To this end, we performed a series of experiments testing whether classification accuracy was affected by use of one-hot probe representations by comparison to embedding probes, in a replication of the word content task of Adi et al. (2016). Finding almost equivalent accuracies between the two input types, we use one-hot probe representations in all subsequent experiments.

Note that as a result, by contrast to Adi et al. (2016) we are not assuming the classifier to identify words in the sentence representation based on resemblance to their original word embeddings—this may not in fact be a safe assumption, given that the word's representation may distort during composition of the sentence. Instead, the classifier must learn a mapping from each one-hot representation to its manifestation in sentences. This means that all words must appear as probes in training. To facilitate the learning of this mapping, we restrict to a small (14-lemma) vocabulary of noun and verb probes in these experiments.[3] Because the generation system is able to produce thousands of sentences from even such a restricted vocabulary as this, this limitation does not prevent generation of adequately large datasets.

A note about the size and selection of the vocabulary: some composition tests will surely be sensitive to specific idiosyncrasies of individual words, in which case the choice of vocabulary will be of great importance. However, for the particular semantic role and negation tasks described here, the focus is on identification of structural dependencies between words, which are not in this case sensitive to the specific nouns/verbs used. Consequently, for these tasks—as long as vocabulary words are not out-of-vocabulary for the models (which we confirm below)—the important thing should be not what the words themselves are, but whether dependencies between them have been captured in the sentence embeddings.

6 Surface tasks: word content and order

Though our ultimate interest is in abstract meaning information, part of the goal of these experiments is to get a clear picture of the information currently captured by existing systems. For this reason, we

[2] More sentences can be found in the publicly available classification datasets.

[3] Sentences themselves contain various morphological inflections of these lemmas.

include the content and order experiments as performed by Adi et al. (2016), to see how encoding of these surface variables compares to encoding of meaning information—and to compare with the results of Adi et al. (2016) after the more rigorous controls used in our datasets.

We structure these tasks to be maximally parallel with our meaning tasks. To this end, we have two content tasks: one-probe ("**Content1Probe**") and two-probe ("**Content2Probe**"), with the one-probe task using verb probes as in the negation task, and two-probe using noun-verb probe pairs, as in the semantic role task. Similarly, for the order task ("**Order**") we use only noun-verb pairs. The order task is thus formulated as "Given representation $\mathbf{n}$ of probe noun n, representation $\mathbf{v}$ of probe verb v, and embedding $\mathbf{s}$ of sentence s (with s containing both n and v), does n occur before v in s?". The two-word content task is formulated as "Given representation $\mathbf{n}$ of probe noun n, representation $\mathbf{v}$ of probe verb v, and embedding $\mathbf{s}$ of sentence s, do both n and v occur in s?", and the one-word content task is formulated as "Given representation $\mathbf{v}$ of probe verb v, and embedding $\mathbf{s}$ of sentence s, does v occur in s?"

7 Classification experiments

To demonstrate the utility of our analysis, we use it to test several existing sentence composition models. Following Adi et al. (2016), for our classifier we use a multi-layer perceptron with ReLU activations and a single hidden layer matching the input size. For each of the above tasks we construct train/test sets consisting of 4000 training items and 1000 test items.[4] No tuning is necessary, as the hyperparameters of hidden layer number and size are fixed in accordance with the architecture used by Adi et al. (2016).

It is important to note that the training of the classifier, which uses the 4000 items mentioned above, is to be distinguished from the training of the sentence embedding methods. The sentence embedding models are pre-trained on separate corpora, as described below, such that they map sentence inputs to embeddings. Once these models are trained, they are used to produce the 4000 sentence embeddings that will serve as training input to the classifier (and the 1000 sentence embeddings used for testing).

Our use of a relatively simple classifier with a single hidden layer builds on the precedent not only of Adi et al. (2016), but also of related methods in neuroscience, which in fact typically use linear classifiers (an option that we could not employ due to our use of the variable probes). An important reason for use of simpler classifiers is to test for *straightforward* extractability of information from embeddings—if a complex classifier is necessary in order to extract the information of interest, then this calls into question the extent to which we might consider this information to be "captured" in the embeddings, as opposed to the information being somehow reconstructable from the embeddings' encoding of other information. That said, the question of how the complexity of the classifier relates to the encoding of the target information in these sentence embeddings is an interesting issue for future work.

For each experiment, we also run two corresponding experiments, in which random vectors are used in place of the sentence vectors and the probes, respectively. This serves as an additional check for biases in the datasets, to ensure that neither the sentence vectors nor the probe vectors alone are sufficient to perform above chance on the tasks. For all tasks, these random vectors produce chance performance.

7.1 Sentence encoding models

We test a number of composition models on these classification tasks. These models represent a range of influential current models designed to produce task-general sentence embeddings. They employ a number of different architectures and objectives, and have shown reasonable success on existing metrics (Hill et al., 2016; Conneau et al., 2017).

All sentence embeddings used are of 2400 dimensions. Because our pre-trained models (SDAE, Skip-Thought) are trained on the Toronto Books Corpus (Zhu et al., 2015), we use this as our default training corpus, except when other supervised training data is required (as in the case of InferSent). Before sentence generation, the chosen vocabulary was checked against the training corpora to ensure that no words were out-of-vocabulary (or below a count of 50).

[4] See footnote 1 for link to all classification datasets used in these experiments.

	Accuracy				
	Content1Probe	Content2Probe	Order	SemRole	Negation
BOW	100.0	97.1	55.0	51.3	50.9
SDAE	100.0	79.8	92.9	63.7	99.0
ST-UNI	100.0	88.1	93.2	62.3	96.6
ST-BI	96.6	79.4	88.7	63.2	74.7
InferSent	100.0	70.1	86.4	50.1	97.2

Table 2: Classification results

BOW averaging Our first sentence embedding model (**"BOW"**) is a BOW averaging model, for which we use the skip-gram architecture of the word2vec model (Mikolov et al., 2013) to learn word embeddings. As discussed above, the BOW model serves primarily as a sanity check for our purposes, but it is important to note that this model has had competitive results on various tasks, and is taken seriously as a sentence representation method for many purposes (Wieting et al., 2016; Arora et al., 2016).

Sequential Denoising Autoencoder Our second model (**"SDAE"**) is an autoencoder variant from Hill et al. (2016) for unsupervised learning of sentence embeddings. The model uses an LSTM-based encoder-decoder framework, and is trained to reconstruct input sentences from their vector representations (last hidden state of encoding LSTM) despite noise applied to the input sentence. We use a pre-trained model provided by the authors. This model has the advantage of an unsupervised objective and no need for sequential sentence data, and it shows competitive performance on a number of evaluations.

Skip-Thought Embeddings Our next two models are variants of the Skip-Thought model (Kiros et al., 2015), in which sentences are encoded with gated recurrent units (GRUs), with an objective of using the current sentence representation to predict the immediately preceding and following sentences. Following the model's authors, we use both the uni-skip (**"ST-UNI"**) and bi-skip (**"ST-BI"**) variants: uni-skip consists of an encoding based on a forward pass of the sentence, while bi-skip consists of a concatenation of encodings of the forward and backward passes of the sentence (each of 1200 dimensions, for 2400 total). We use the publicly available pre-trained Skip-Thought model for both of these variants.[5]

Skip-Thought sentence embeddings have been used as pre-trained embeddings for a variety of tasks. They have proven to be generally effective for supervised tasks and passable for unsupervised tasks (Hill et al., 2016; Triantafillou et al., 2016; Wieting et al., 2016). Like the SDAE model, the Skip-Thought model is able to use unsupervised learning, though it requires sequential sentence data. However, more than the SDAE model, the Skip-Thought model uses an objective intended to capture semantic and syntactic properties, under the authors' assumption that prediction of adjacent sentences will encourage more syntactically and semantically similar sentences to map to similar embeddings.

InferSent Our final model is the InferSent model (Conneau et al., 2017), which uses multi-layer BiLSTM encoders with max pooling on the hidden states of the last layer to produce vector representations of the sentences. This model is trained with a natural language inference (NLI) objective, and for this reason we train it on the SNLI dataset (Bowman et al., 2015).

The InferSent model is intended to produce "universal" sentence representations, and has been shown to outperform unsupervised methods like Skip-Thought on a number of tasks (Conneau et al., 2017). More generally, the NLI objective is believed to encourage learning of compositional meaning information, given that inference of entailment relations should require access to meaning information.

7.2 Results and Discussion

Table 2 shows the accuracy of the different models' sentence embeddings on our classification tasks.

The first thing to note is that our BOW control allows us to confirm nearly complete lexical balance in

[5] https://github.com/ryankiros/skip-thoughts

the sentence sets: the averaged word embeddings perform roughly at chance on all but the content tasks.[6] By contrast, BOW performs with near-perfect accuracy on the content tasks, lending support to the intuitive conclusion: the one thing that BOW *does* encode is word content. The quality of performance of the BOW model on this task exceeds that reported by Adi et al. (2016)—we speculate that this may be due to our use of a smaller vocabulary to facilitate the learning of the mapping from one-hot probes.

While BOW has very high performance on two-probe word content, SDAE, ST-UNI, ST-BI and InferSent have much lower accuracy (albeit still far above chance), suggesting that some detail with respect to word content is sacrificed from these representations in favor of other information types. This is exemplified by the order task, on which all non-BOW models show significantly higher accuracy than on the word content tasks, supporting the intuitive conclusion that such sequence-based models retain information about relative word position. This result is generally consistent with the Adi et al. (2016) result, but due to the additional control that brings BOW roughly to chance, we can conclude with greater confidence that the performance on this task pertains to order information in the source sentence itself.

Turning to our meaning information tasks, we see that with the exception of ST-BI, the sequence models perform surprisingly well on the negation task, despite the fact that this task cannot be solved simply by detecting adjacency between negation and the verb (due to our insertion of adverbs). Instead, we speculate that these sequence models may be picking up on the utility of establishing a dependency between negation and the *next* verb, even in the face of intervening words. This is not a complete solution to the problem of representing the meaning and dependencies of negation, but it is a useful step in that direction, and suggests that models may be sensitive to some of the behaviors of negation.

Interestingly, ST-BI shows markedly weaker performance on the negation task. We see two potential reasons for this. First, it may be due to the reduced dimensionality of each of the two concatenated encodings (recall that ST-BI involves concatenating 1200-dimensional encodings of the forward and backward passes). Second, the reduced performance could be influenced by the inclusion of the backward pass: while the forward pass can leverage the strategy of linking negation to the next verb, the backward pass cannot use this strategy because it will encounter the relevant verb before encountering the negation.

Turning to the semantic role task, we see a stark contrast with the high performance for the negation task. InferSent performs squarely at chance, suggesting that it retains as little compositional semantic role information as does BOW. SDAE, ST-UNI and ST-BI perform modestly above chance on the semantic role task at 62-63% accuracy, suggesting that they may provide some amount of abstract role information—but no model shows any substantial ability to capture semantic role systematically.

These results accomplish two things. First, they lend credence to this method as a means of gaining insight into the information captured by current models. Second, they give us a sense of the current capacity of sequence-based models to capture compositional meaning information. The picture that emerges is that sequence models are able to make non-trivial headway in handling negation, presumably based on a sequential strategy of linking negation to the next verb—but that these sequence models fall significantly short when it comes to capturing semantic role compositionally. Another point that emerges from these results is that despite the fairly substantial differences in architecture, objective, and training of these models, capacity to capture the compositional information is fairly similar across models, suggesting that these distinct design decisions are not having a very significant impact on compositional meaning extraction. We plan to test more substantially distinct models, like those with explicit incorporation of syntactic structure (Bowman et al., 2016; Dyer et al., 2016; Socher et al., 2013) in future work.

8 Related work

This work relates closely to a growing effort to increase interpretability of neural network models in NLP—including use of visualization to analyze what neural networks learn (Li et al., 2015; Kádár et al., 2016), efforts to increase interpretability by generating explanations of model predictions (Ribeiro et al., 2016; Lei et al., 2016; Li et al., 2016), and work submitting adversarial examples to systems in order to identify weaknesses (Zhao et al., 2017; Jia and Liang, 2017; Ettinger et al., 2017).

[6]The slightly higher accuracy on the order task is most likely the result of a very slight bias due to our use of only noun-verb order probe pairs for the sake of matching the SemRole task.

Methodologically the most closely related work is that of Adi et al. (2016), which uses classification tasks to probe for information in sentence embeddings. As discussed above, we depart from that work in targeting deeper and more linguistically-motivated aspects of sentence meaning, and we incorporate careful controls of our datasets to ensure elimination of bias in the results.

Our focus on assessing linguistically-motivated information relates to work on evaluations that aim for fine-grained analysis of systems' linguistic capacities (Rimell et al., 2009; Bender et al., 2011; Marelli et al., 2014). The present work contributes to this effort with new tasks that assess composition *per se*, and that do so in a highly targeted manner via careful controls. Our use of synthetically generated data to achieve this level of control relates to work like that of Weston et al. (2015), which introduces synthetic question-answering tasks for evaluating the capacity of systems to reason with natural language input.

Our examination of the capacity of neural sequence models to identify abstract relations in sentence representations also relates to work by Linzen et al. (2016), who explore whether LSTMs can learn syntactic dependencies, as well as Williams et al. (2017), who investigate the extent to which parsers that are learned based on a semantic objective produce conventional syntax.

Finally, importantly related work is that concerned specifically with testing systematic composition. Lake and Baroni (2017) investigate the capacity of RNNs to perform zero-shot generalization using composition, and Dasgupta et al. (2018) construct an entailment dataset with balanced lexical content in order to target composition more effectively. We contribute to this line of inquiry by establishing an analysis method that can take output embeddings from sentence composition models and query them directly for specific types of information to be expected in properly compositional sentence representations.

9 Conclusions and future directions

We have presented an analysis method and accompanying generation system designed to address the problem of assessing compositional meaning content in sentence vector representations. We make the datasets for these tasks, as well as the generation system used to create them, available for public use to facilitate broader testing of composition models. We have also presented the results of applying this method for analysis of a number of current sentence composition models, demonstrating the capacity of the method to derive meaningful information about what is captured in these models' outputs.

Having established a means of analyzing compositional meaning information in sentence embeddings, in future work we plan to apply this system to identify more precisely which design decisions lead to effective capturing of meaning information, in order to guide system improvement. As part of this effort, we will expand to more comprehensive testing of a diverse range of sentence embedding systems (Bowman et al., 2016; Subramanian et al., 2018). We also plan to investigate the potential of our generation system to create not just evaluation data, but training data—given that it allows us to produce large, meaning-annotated corpora. Finally, we plan to expand beyond semantic role and negation in the set of information types targeted by our method, in order to establish more comprehensive coverage of meaning information that can be assessed by this analysis system.

Acknowledgements

Devin Ganey contributed code interfacing the generation system and the XTAG database. The work described in this paper benefited from discussions with Alexander Williams, Marine Carpuat, and Hal Daumé III, and from helpful comments by Ido Dagan, Jason Eisner, Chris Dyer, audience members at RepEval 2016, members of the UMD CLIP and CNL labs, as well as anonymous reviewers. This work was supported in part by an NSF Graduate Research Fellowship to Allyson Ettinger under Grant No. DGE 1322106, and by NSF NRT Grant DGE-1449815. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the NSF.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2012. Abstract meaning representation (AMR) 1.0 specification. In *Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL*, pages 1533–1544.

Emily M. Bender, Dan Flickinger, Stephan Oepen, and Yi Zhang. 2011. Parser evaluation over local and non-local deep dependencies in a large corpus. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 397–408, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Luisa Bentivogli, Raffaella Bernardi, Marco Marelli, Stefano Menini, Marco Baroni, and Roberto Zamparelli. 2016. SICK through the SemEval glasses. Lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *Language Resources and Evaluation*, 50(1):95–124.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. 2018. Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*.

Christy Doran, Dania Egedi, Beth Ann Hockey, Bangalore Srinivas, and Martin Zaidel. 1994. XTAG system: a wide coverage grammar for English. In *Proceedings of the 15th conference on Computational linguistics-Volume 2*, pages 922–928. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent neural network grammars. *NAACL*.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP*, page 134.

Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M Bender. 2017. Towards linguistically generalizable NLP systems: A workshop and shared task. *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *NAACL*.

James V Haxby, Andrew C Connolly, and J Swaroop Guntupalli. 2014. Decoding neural representational spaces using multivariate pattern analysis. *Annual review of neuroscience*, 37:435–456.

Irene Heim and Angelika Kratzer. 1998. *Semantics in generative grammar*, volume 13. Blackwell Oxford.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *NAACL*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Ákos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2016. Representation of linguistic form and function in recurrent neural networks. *arXiv preprint arXiv:1602.08952*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284.

Brenden M Lake and Marco Baroni. 2017. Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks. *arXiv preprint arXiv:1711.00350*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in NLP. *arXiv preprint arXiv:1506.01066*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Language Resources and Evaluation*, pages 216–223.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

M Palmer, D Gildea, and P Kingsbury. 2005. The Proposition Bank: A corpus annotated with semantic roles. *Computational Linguistics*, 31(1):712105.

Ehud Reiter, Robert Dale, and Zhiwei Feng. 2000. *Building natural language generation systems*, volume 33. MIT Press.

Philip Resnik. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61(1-2):127–159.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 813–821, Singapore. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *ICLR*.

Eleni Triantafillou, Jamie Ryan Kiros, Raquel Urtasun, and Richard Zemel. 2016. Towards generalizable sentence embeddings. In *ACL Workshop on Representation Learning for NLP*, page 239.

Tim Van de Cruys. 2014. A neural network approach to selectional preference acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 26–35.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards AI-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *International Conference on Learning Representations (ICLR)*.

Adina Williams, Andrew Drozdov, and Samuel R Bowman. 2017. Learning to parse from a semantic objective: It works. is it syntax? *arXiv preprint arXiv:1709.01121*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2017. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.