Frame Identification as Categorization: Exemplars vs Prototypes in Embeddingland

Jennifer Sikos
IMS, University of Stuttgart
Stuttgart, Germany
jen.sikos@ims.uni-stuttgart.de

Sebastian Padó
IMS, University of Stuttgart
Stuttgart, Germany
pado@ims.uni-stuttgart.de

Abstract

Categorization is a central capability of human cognition, and a number of theories have been developed to account for properties of categorization. Despite the fact that many semantic tasks involve categorization, theories of categorization do not play a major role in contemporary research in computational linguistics. This paper follows the idea that embedding-based models of semantics lend themselves well to being formulated in terms of classical categorization theories. The benefit is a group of models that enables (a) the formulation of hypotheses about the impact of major design decisions, and (b) a transparent assessment of these decisions.

We instantiate this idea on the frame-semantic frame identification task. We define four models that cross two design variables: (a) the choice of prototype vs. exemplar categorization, corresponding to different degrees of *generalization* applied to the input, and (b) the presence vs. absence of a fine-tuning step, corresponding to generic vs. task-adaptive categorization. We find that for frame identification, generalization and task-adaptive categorization both yield substantial benefits. Our prototype-based, fine-tuned model, which combines the best choices over these variables, establishes a new state-of-the-art in frame identification.

1 Introduction

Categorization is the process of forming categories and assigning objects to them, and is a central capability of human cognition (Murphy, 2002). Not surprisingly, cognitive psychology has shown substantial interest in theories of categorization. Two such prominent theories are *prototype* and *exemplar* models. In prototype theory, categories are characterized in terms of a single representation, the prototype, which is an abstraction over individual objects and captures the ‘essence’ of the category (Posner and Keele, 1968; Rosch, 1975). In computational models, the prototype is often computed as the centroid of the objects of a category, and new objects are classified by their similarity to different categories’ prototypes. As a result, the decision boundary between every pair of categories is linear. In contrast, exemplar theories represent categories in terms of the potentially large set of objects, called exemplars, that instantiate the category (Nosofsky, 1986; Hintzman, 1986). New objects are classified by similarity to nearby exemplars, so in a computational model this becomes similar to a nearest-neighbor classification. In exemplar models, the decision boundary between categories can become non-linear, enabling more complex behavior to be captured, but at the cost of higher training data requirements.

Prototype and exemplar theories are typically not at the center of attention in contemporary computational linguistics. One reason is arguably that, due to their origin in psychology, they tend to restrict themselves to cognitively plausible parameters and learning mechanisms (Nosofsky and Zaki, 2002; Lieto et al., 2017), whereas the focus of computational linguistics is very much on the use of novel machine learning techniques for applications. We nevertheless believe that categorization theory is still relevant for computational linguistics, and lexical semantics in particular. In fact, the emergence of distributed representations (embeddings) as a dominant representational paradigm has had a unifying effect on work in lexical semantics. The properties of high-dimensional embeddings provide a good match with the assumption behind much of categorization theory – namely, that categories arise naturally from the similarity structure of individual objects (Erk, 2009).

Given this context, the exemplar–prototype dichotomy is a useful dimension on which models can be situated in terms of how much they generalize over objects: low for exemplar-inspired, but high for prototype-inspired models. Regarding the representation of word meaning in context, for example, the additive models first considered by Mitchell and Lapata (2008) fall into the prototype camp, while Erk and Padó (2010) propose exemplar-based models, and Reisinger and Mooney (2010) explore dynamic generalization in what they called 'multi-prototype' categorization models. However, for many tasks, such comparisons – on a level playing field – are missing.

An interesting recent development in the embedding literature is the emergence of the distinction between *pre-training* and *fine-tuning* (e.g., in BERT (Devlin et al., 2018), OpenAI's GPT (Radford et al., 2018), or ULM (Howard and Ruder, 2018)): pre-training constructs embeddings that are supposedly general and are robust across many tasks. Fine-tuning can then further optimize embeddings for one particular task, at the cost of robustness. Importantly, pre-training takes advantage of massive amounts of unlabeled data, while fine-tuning can leverage small amounts of task-specific labeled data. This distinction ties in nicely with open questions in the categorization literature concerning the respective roles of "bottom-up" similarity information and "top-down" theory information (Smith and Sloman, 1996): task-independent pre-training embeddings, and their similarities which shape the categorization process, can be understood as "bottom-up" information, while the transformations that fine-tuning introduces to optimize these embeddings for a specific task, arguably represent "top-down" information. Notably, such transformations can be understood equivalently as learning task-specific similarity metrics (Bellet et al., 2013). By learning general representations in a bottom-up pre-training phase and then comparing performance with additional top-down fine-tuning, we can discriminate how much general semantic knowledge is necessary to perform a categorization task and how much task-specific learning is required.

In this paper, we investigate a lexical semantic task, specifically the identification of frame-semantic frames (Baker et al., 1998) in running text, from this categorization perspective. Frames can be understood as semantic classes that are sensitive both to the topic of the context and to specific properties of the predicate-argument structure. We present four categorization models for the task, all of which are based on the state-of-the-art BERT model (Devlin et al., 2018) but which differ in how they use its embeddings. Two models are prototype-based (i.e., compute a representation for each frame), and two are exemplar-based (i.e., represent a frame solely in terms of its instances). Within each group, we compare the use of embeddings without fine-tuning ("bottom-up") and with fine-tuning ("top-down").

Contributions and Findings. This setup enables us to gauge, on a lexical semantic analysis task, (a) whether generalization helps, and what the size of the effect is; (b) whether there are benefits of top-down task adaptation; (c) whether there is an interaction between generalization and adaptation. We find that generalization indeed helps, as does top-down adaptation. Overall, our best model establishes a new state-of-the-art in frame identification.

Structure of the paper. In Section 2, we provide details on frame semantics and frame identification, as well as the current work in distributed semantic representations. We additionally outline the architecture of BERT its pre-training and fine-tuning steps. Section 3 defines the four models that we experiment with, and Section 4 describes the experimental setup. Finally, we describe and discuss results and analysis in Sections 5 and 6.

2 Background

2.1 Frame Semantics and Frame Identification

Frame Semantics is a theory of semantics that groups predicates in terms of the situations that they describe and their relevant participants (Fillmore, 1982). These situations, or scenarios, are formalized in terms of *frames*, conceptual categories which have a set of *lexical units* that evoke the situation, and a set of *frame elements* that categorize the participants and that are expected to be realized linguistically. For instance, *tell*, *explain*, and *say* are all capable of expressing the STATEMENT frame which describes the situation where SPEAKER communicates a MESSAGE to a RECIPIENT.

Frame Semantics has been implemented in a number of NLP applications thanks to the Berkeley FrameNet resource (Baker et al., 1998). The latest FrameNet lexicon release provides definitions for over 1.2k frames, and 13,640 lexical units (i.e., predicate–frame pairs), where there are approximately 12 lexical units per frame. FrameNet also provides sentence annotations that mark, for each lexical unit, the frame that is evoked as well as its frame elements in running text. This annotated corpus has sparked a lot of interest in computational linguistics, and the prediction of frame-semantic structures (frames and frame elements) has become known as *(frame-)semantic parsing* (Gildea and Jurafsky, 2002; Das et al., 2014).

2.1.1 Frame Identification

In this paper, we focus on the first step of frame-semantic parsing called *frame identification* or *frame assignment*, where an occurrence of a predicate in context is labeled with its FrameNet frame. This is a categorization task that presents two main challenges:

Ambiguity Most predicates in FrameNet are ambiguous, that is, they can evoke different frames depending on the context that they occur in. For example, *treat* has a medical sense (*treat a disease*) that evokes the MEDICAL_INTERVENTION frame and a social sense (*treat a person in some manner*) that evokes the TREATING_AND_MISTREATING frame. These distinctions can be relatively subtle: *say* can evoke (among others) the frames STATEMENT and TEXT_CREATION which differ mainly in the modality of the communicative act (*said to his friend* vs. *said in his book*).

Generalization As conceptual categories, frames are clearly open classes. No resource can exhaustively list all frames or the predicates that can evoke them.

Frame identification was first modeled as a supervised classification task, based on linguistic features (Das et al., 2010). While such systems address the ambiguity problem to some degree, they tend to struggle with generalization. An alternative approach investigated the use of other machine-readable dictionaries (Green et al., 2004), but was not able to fundamentally overcome the generalization problem.

Recent progress in supervised frame identification has come out of neural networks and distributed word representations (Peng et al., 2018; Hartmann et al., 2017). In these studies, frame-labeled corpora are used to learn embeddings for the frames as a side product of representation learning with different objective functions. Hermann et al. (2014) learned embeddings jointly for frames and the sentential contexts in which they were evoked. The current state-of-the-art in frame identification performs full-fledged semantic role labeling, i.e., it jointly assigns frames as well as frame elements, using a bi-directional LSTM architecture (Peng et al., 2018).

2.2 Distributed Representations of Word Meaning

Distributed representations of word meanings (embeddings) have become a standard representation format in semantics. These models are grounded in the distributional hypothesis (Harris, 1954), according to which similar words are expected to appear in similar contexts. Based on this hypothesis, word (and phrase) meaning is represented as vectors (embeddings) in a semantic space whose dimensions correlate with properties of the context, and in which closeness between two vectors indicates semantic relatedness.

Traditionally, "count" vectors were created by simply counting co-occurring context features, with the option of additional weighting and compression over those count vectors. Neural network-based "predict" vectors are learned by treating contextual features as parameters of an objective function that is optimized on a corpus. One of the first, and still popular, "predict" models is the word2vec Skipgram model (Mikolov et al., 2013). It optimizes a word embedding using a context bag of words. This model, however, learns representations only at the lexical level, so that occurrences of a word in different contexts (cf. *treat* in Section 2.1.1) are represented equally. This has changed with the latest generation of embedding models, such as AllenNLP's ELMo (Gardner et al., 2018) and Google's BERT (Devlin et al., 2018), which build *contextualized embeddings* for occurrences of words based on the context as well as their relative positions.

A second important recent development concerns the objective(s) used to learn the embedding. While traditional count vectors and early embedding models like Skipgram assume that embeddings are general, and trained in an *task-agnostic* fashion, there is an alternative thread of research that sees the training of embeddings as a side product of training *task-specific* neural network models on tasks like sentiment analysis or machine translation (Socher et al., 2013; Hill et al., 2017). The most recent models reconcile these two directions with a two-phase transfer learning setup. The first phase is *pre-training*, where task-agnostic embeddings are learned from large, unlabeled corpora. The second phase is *fine-tuning*, which adapts the pre-trained embeddings to a specific task using comparatively small amounts of task-specific labeled corpora.

2.2.1 Bidirectional Encoder Representations from Transformers (BERT)

BERT (Devlin et al., 2018) is a state-of-the-art embedding model that provides contextualized embeddings in a pre-training/fine-tuning setup. BERT is essentially a deep network of Transformer blocks (Vaswani et al., 2017) which use stacked self-attention mechanisms to capture relationships across different positions in a sequence. The two tasks that are used for pre-training are language modeling and recognition of discourse continuation. Representations from the pre-training step are then pooled and fed to the fine-tuning stage for classification. Fine-tuning proceeds by adding an additional, task-specific layer on top of the pre-trained BERT embeddings that maps embeddings onto the desired task output. In addition to learning weights for this final, task-specific classification layer, this procedure also updates the pooled, pre-trained embedding through backpropagation.

3 Categorization Approaches to Frame Identification

This section defines our four embedding-based models for frame identification. As motivated in Section 1, we based our model space on two well-known dichotomies from categorization research: exemplar vs. prototype theory, and pure bottom-up processing vs. a combined bottom-up plus top-down processing. This setup results in a 2x2 matrix and a total of four models, as sketched in Figure 1. To focus solely on the problem of frame identification as a categorization task, we assume that the frame-evoking predicates have already been identified in the texts of interest.

The first dimension distinguishes prototype vs exemplar models, shown in the figure as columns. We consider models to be exemplar-based when they only use representations for individual instances for their predictions (here, predicates in context), but do not compute aggregate embeddings at the category level (here, frames). One of the most straightforward implementations of this approach is nearest-neighbor classification (Daelemans and van den Bosch, 2005). In contrast, prototype-models do not use instance representations at prediction time, but instead aggregate them into category-level representations. The standard softmax classification, for example, is a clear prototype approach by virtue of learning a weight vector for each class whose dot product with an input represents the probability of that class. The geometric interpretation of this computation is a distance between the prototype vector and the input, using dot product.

The second dimension, shown in the figure as rows, distinguishes pure bottom-up from bottom-up plus top-down models. In categorization research, bottom-up models assume that general similarity information "as given" is sufficient to perform the classification. In an embedding-based setup, this corresponds to models where embeddings are (pre-)trained in a task-independent fashion and applied to the task as-is. Thus, the bottom-up models form categories purely from contextual features that have been learned in a generalized, unsupervised fashion. In contrast, combined bottom-up plus top-down models assume that top-down information, such as a preconceived notion of a category or similarity

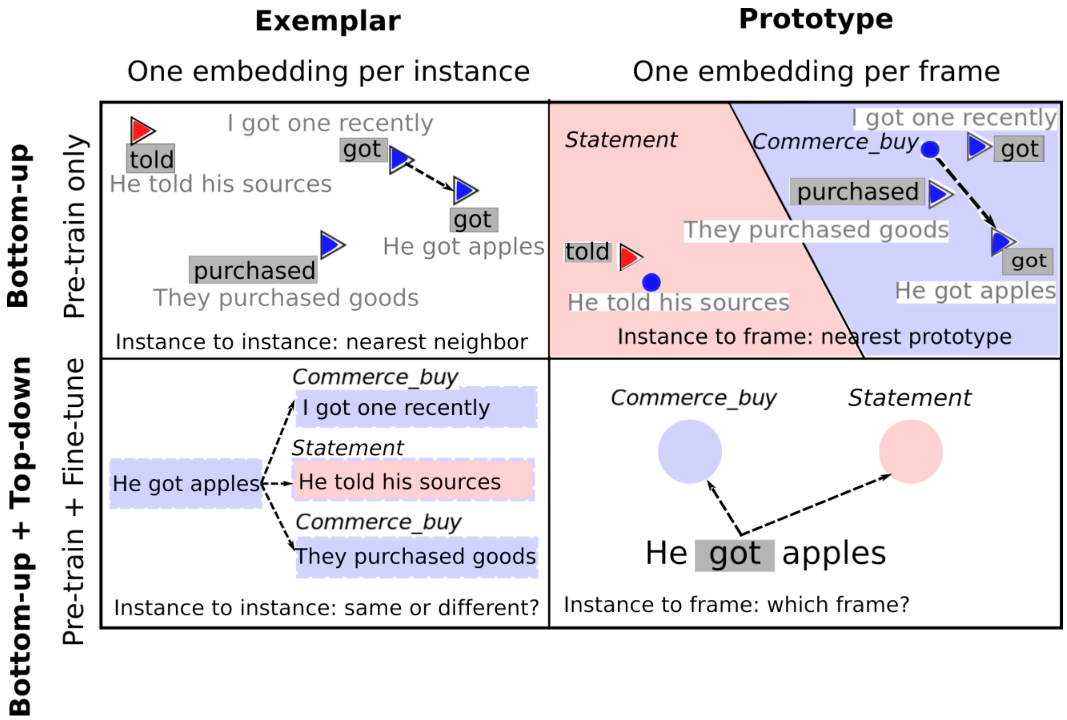

Figure 1: Four categorization models for frame identification, showing processing of the same predicate in context (*He got apples*) across model architectures. Blue stands for frame COMMERCE_BUY, light red for frame STATEMENT. Triangles are instances, dots are prototypes.

metric, influences processing for this task. In the context of current embedding-based models, we treat the fine-tuning procedure in BERT (cf. Section 2.2.1), where representations are fine-tuned using a small amount of task-specific data as an approximate top-down effect on categorization.

3.1 Bottom-up (Pre-trained Embeddings)

Bottom-up frame identification models use only the pre-trained embeddings to predict the frame of a lexical unit in context. The classification performed by these models shows how well frame classification can be carried out by relying on general lexical semantic relatedness, without explicit knowledge about frame-semantic grouping.

Bottom-up Exemplar. In exemplar theories, categorization proceeds by comparing a target instance to prior seen instances, and the target is assigned the same class as its closest seen instance. To classify a predicate in context, we perform single nearest neighbor classification: we compare its pre-trained, contextualized embedding to all pre-trained, contextualized embeddings of predicates in the training set, and assign the frame label of the closest training predicate. We use the standard embedding similarity metric, cosine similarity. In the example in Figure 1 (top left), the nearest neighbor to the test instance *He got apples* is *I got one recently*, which leads to the assignment of the COMMERCE_BUY frame.

Bottom-up Prototype. In the prototype model, the frame categories are formed by building a summary representation of all known instances in a category. We take advantage of the general effectiveness of averaged representations and compute frame prototypes as the unweighted centroid of all pre-trained, contextualized predicate embeddings for the frame's training instances. Frame classification then assigns a novel instance to the category of its most similar prototype. We again use cosine similarity, which is identical (modulo normalization) to softmax classification. The example in Figure 1 (top right) shows the prototypes of the two frames as dots, the "regions" of the two frames by background color, as well as the (linear) decision boundary between prototypes. The test instance *He got apples* is assigned to the COMMERCE_BUY frame because it is closest to the prototype of that frame.

3.2 Combined Bottom-up plus Top-down (Pre-trained and Fine-tuned Embeddings)

Bottom-up plus top-down frame identification models optimize the embeddings according to task-specific data and a loss function during a fine-tuning phase. Prediction then uses the fine-tuned embeddings instead of pre-trained ones.

Bottom-up plus Top-down Exemplar. For our exemplar-based model, we apply fine-tuning to make the embeddings for predicates that evoke the same frame more similar, and embeddings evoking different frames less similar. We frame the fine-tuning step as a binary classification task that decides, for a pair of predicates in sentence context, whether they evoke the same or different frames. The input consists of the concatenated contextualized embeddings of the two predicates. An example of the training input is below, where the BERT model adds a special [SEP] token between the text pair. The *treated* predicate from the first sentence evokes the MEDICAL_INTERVENTION frame, whereas the *got* predicate in the second sentence evokes COMMERCE_BUY.

Input Sequence	The doctor treated the patient [SEP] He got apples
Label	different

Formally, for each predicate in context i with frame $f(i)$, we define $P^+(i)$ as a set of (positive) instances with the same frame, $P^-(i)$ as a set of (negative) instances with different frames, and $\mathit{same}(i, i')$ as the binary prediction of the model. We then define the objective function as a cross-entropy loss between the gold label (same / different) and $\mathit{same}(i, i')$.

$$L_{ex} = -\sum_{i} [\sum_{i' \in P^+(i)} \log p(\mathit{same}(i, i')) + \sum_{i' \in P^-(i)} \log(1 - p(\mathit{same}(i, i')))] \quad (1)$$

We select $P^-(i)$ from the set of frame candidates for a given lexical unit. For predicates with only a single frame, we randomly select a negative instance from the entire frame class inventory. For each predicate in the training data, we use two positive and negative instances, which we obtain by random sampling.

At prediction time, we pair the target predicate with all instances of all frame candidates for this predicate and run them through the trained classification model, as shown in the bottom-left corner of Figure 1. For Unseen predicates (see Section 4), we pair target predicates with one randomly selected example from each frame in the entire frame inventory. We then label the target predicate with the frame of the instance with the highest same-frame probability. In this model, the top-down knowledge that is passed to the network corresponds to the similarity metric between frame-evoking predicates.

Bottom-up plus Top-down Prototype. For the prototype model, we fine-tune the embeddings specifically to learn frame classes (cf. the bottom right hand example in Figure 1). Since we will train on full-text annotation (described in further detail in Section 4), frame identification proceeds as a token sequence classification, where each token is assigned a frame prediction. An example of the training data is shown below, where the input to the model is the sequence of plain text tokens, and the gold class labels are the sequence of correct frame assignments. In the gold label sequence, non-predicates are assigned an outside (O) label.

Input Sequence	The	doctor	treated	the	patient
Label	O	MEDICAL_PROFESSIONALS	MEDICAL_INTERVENTION	O	MEDICAL_INTERACTION_SCENARIO

The loss function is a straightforward multi-class cross-entropy loss averaged over each class for every token. Here, the set of labels are the entire set of frames in the FrameNet lexicon, plus the added ‘outside’ class label – resulting in a large set of possible classes (1,021 classes). At prediction time, the model predicts a frame label for each token in the input sequence independently. As is the case in the bottom-up prototype model, no global optimization takes place. We only consider predictions for predicates (according to the gold standard) for the purposes of evaluation.

4 Experiments

4.1 Dataset

We work with the dataset sampled by Das et al. (2014) from the FrameNet Release 1.5 full-text annotations. This dataset contains a total of 78 documents with frame-annotated sentences drawn from the British National Corpus. In total, 39 documents were selected for training and 16 for development with a total of 19,582 target predicates, and 23 documents for testing with 4,458 target predicate annotations. This is the standard dataset used for evaluation of frame identification systems.

4.2 Model Setup and Hyperparameters

BERT provides several pre-trained models for English that were trained on the concatenation of the BooksCorpus (Zhu et al., 2015) and Wikipedia. We use the pre-trained BERT-large, cased model, trained with the highest number of layers (L=24), hidden units (H=1024), and self-attention heads (A=16). The final layer of the BERT transformer provides embeddings for each token in the sentence that can be interpreted as contextualized meaning representations. According to the authors of the BERT model, performance is shown to improve when the n final layers for each token are concatenated. We use n=4.

For the fine-tuned models, we re-used the hyperparameters of the pre-trained model. Since both of our fine-tuning tasks are classification tasks, we add a standard softmax classification layer with cross-entropy loss on top of BERT (described earlier in Section 3.2). Due to the computational cost of attention mechanisms, the fine-tuned models require a limit on the maximum sequence length. We set the sequence length to 180 in the prototype model, which in this case means that even long sentences can be fed to the model. The exemplar model, on the other hand, takes two text sequences as input (see Section 3.1) which doubles the overall size of the input sequence. We increased the maximum sequence length to 200 for this model to keep as many tokens as possible in training while also being computationally feasible.

We note that our prototype model required a significant number of epochs to converge. Most tasks in the BERT paper achieve near-optimal accuracy with 3–4 training epochs, while our model required about 30 epochs. We attribute this to the number of classes (at most 4 classes in the BERT paper, more than 1000 classes for frame identification). The exemplar model follows more closely with other BERT tasks, and we perform 5 training epochs for the exemplar model.

4.3 Evaluation Metrics

The general evaluation metric for frame identification is accuracy: the relative frequency of correct assignments to predicates. Since the task of frame identification is moot for single-frame lexical units, frame identification systems standardly (Das et al., 2014; Peng et al., 2018; Hermann et al., 2014) report accuracy on two different subsets of the data: (1) all instances from the test set, called “Full Lexicon”, because it includes lexical units that are unambiguous; and (2) only instances of predicates from the test set that can evoke multiple frames, called “Ambiguous”. In the data set we use, the test partition contains 2,029 ambiguous predicates out of a total of 4,458 predicate instances.

In addition, some prior work reports specific metrics on infrequent predicates, for which prediction is particularly challenging. “Unseen” reports accuracy for predicates that are completely unseen in the training data and their predictions over all possible frames – meaning the frame lexicon is not used for evaluation at test time[1]. “Rare” reports accuracy on predicates that occur less than 11 times in the training

[1] (Das et al., 2014) improve their Unseen results with a graph that was constructed over a large corpus of sentences in combination with the FrameNet lexical unit example sentences. We only report Unseen results which where produced over the

	Model	Full Lexicon	Ambiguous	Rare	Unseen
Results from literature	Das et al. (2014)	83.60	69.19	82.31	23.08
	Hermann et al. (2014)	88.41	73.10	85.04	44.67
	Hartmann et al. (2017)	87.63	73.8	NA	NA
	Yang and Mitchell (2017)	88.2	75.7	NA	NA
	Peng et al. (2018)	90.0	78.0	NA	NA
	Model	**Full Lexicon**	**Ambiguous**	**Rare**	**Unseen**
Our Work	Bottom-up Exemplar	82.52	64.44	81.09	11.07
	Bottom-up Prototype	84.67	69.18	83.68	09.59
	Bottom-up + Top-down Exemplar	84.09	65.06	84.18	18.89
	Bottom-up + Top-down Prototype	**91.26**	**80.77**	**91.85**	30.20

Table 1: Accuracy results for Frame Identification on Das et al. (2014) benchmark dataset (test partition)

data. The test set contains 144 unseen and 2,555 rare predicates.

5 Results

Table 1 shows the performance of the four models as well as prior results from recent literature. Regarding the impact of the exemplar and prototype dimensions that we introduced in Section 3, we find that the exemplar model does worse overall than the prototype model in both configurations (overall "Full Lexicon" accuracy: 2% for bottom-up, 7% for bottom-up plus top-down). This indicates that the prototype setup appears better suited to the task than the exemplar one, at least on the data we experimented with. Second, we see a substantial effect of top-down processing (fine-tuning): 1.5% for exemplars, over 6% for prototypes. The clear winner is the bottom-up plus top-down (fine-tuned) prototype model: with an accuracy of 91.26%, it outperforms the previous state of the art (Peng et al., 2018). This shows that frame categorization can indeed profit from task-based optimization. That being said, it is worth noting that even the bottom-up prototype model with only generic pre-training performs at or above the level of the supervised SEMAFOR model (Das et al., 2014) which incorporated linguistic and ontological features in a log-linear model. Thus, the bottom-up vector space models do have a claim to robust performance.

Accuracy on "Ambiguous" predicates largely mirrors the patterns we find on "Full Lexicon" accuracy. They bolster the interpretation that both prototype representation and fine-tuning lead to clear gains. Results on "Rare" and "Unseen" predicates are more difficult to compare due to lack of reported results (marked as NA). The numbers for "Rare", again, seem to follow the "Full Lexicon" trend, and outperform the state of the art. The results for the "Unseen" category do so too, but are below the previously reported results. The reason is that Das et al. (2014) employ additional processing to unseen predicates based on a context similarity graph. For simple supervised classification without the extra component, comparable to our 30.20% setting, they report an Accuracy of 23.08%.

5.1 Sentence Length

Next, we aim to determine how much the sentence length affects predictions of classes in the bottom-up versus the bottom-up plus top-down models. Results are shown in Figure 2. We find that the performance of the bottom-up models declines as sentence length increases, and the opposite is seen in the top-down prototype model.

The most natural explanation for this pattern starts from the realization that the BERT model incorporates long-range dependencies via its self-attention mechanisms. That is, these long-range dependencies, coupled with the bidirectionality in the BERT model, introduces a rich notion of context. However, in the

full-text annotations for fair comparison.

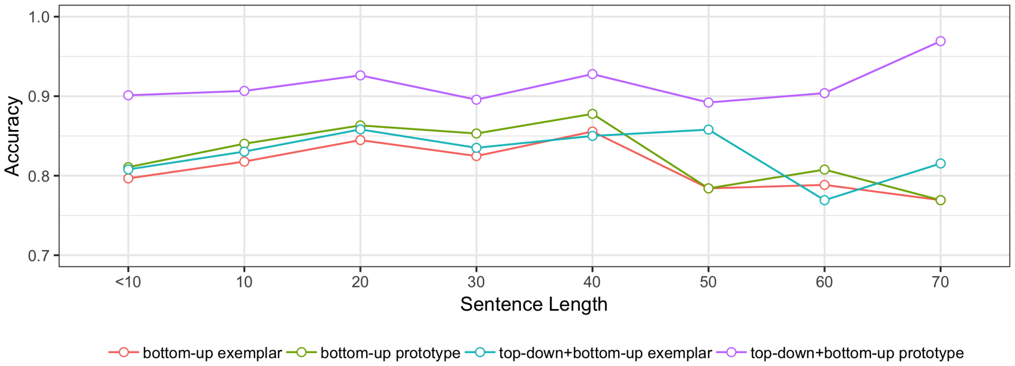

Figure 2: Impact of sentence length on accuracy

Frame	**BU+TD Prototype**	BU+TD Exemplar	BU Prototype	BU Exemplar
CAPABILITY	1.00	0.73	0.48	0.73
POSSESSION	1.00	0.94	0.92	0.81
WEAPON	1.00	0.97	0.98	1.00
LOCATIVE_RELATION	0.97	0.84	0.89	0.79
TEMPORAL_COLLOCATION	0.89	0.76	0.76	0.71

Table 2: Accuracies for top 5 frames from Bottom-up+Top-down Prototype model across all four model

bottom-up models these self-attention weights have the potential to introduce noise for long sentences, which is exactly what we observe. In contrast, fine-tuning of the self-attention weights can apparently turn long sentences into an asset by providing rich context hints for improved frame classification.

The outlier in this analysis is the fine-tuned bottom-up plus top-down exemplar model whose performance fluctuates between the fine-tuned prototype model and the bottom-up models. Given the analysis of the previous paragraph, this may not be surprising: the supervision provided to the fine-tuned exemplar model is less informative than that for the prototype model (cf. Section 3.2): the exemplar supervision does not name the frame(s) involved, and only provides information for one predicate pair in a potentially long sequence. Arguably, this makes it much more difficult for BERT to properly adapt its self-attention weights.

5.2 Frame-level and Predicate-level Analysis

We now look at the most accurate frames and predicates from our best model and compare the accuracies for these inputs across our four models. This analysis gives us insight regarding what types of semantic information are already learned by the bottom-up models versus the knowledge that is gained by learning frame-specific semantics in the top-down setting.

Table 2 shows the analysis at the frame level. The best model assigns three frames perfectly. For one of them, CAPABILITY, there is a dramatic performance gap, where the other models show accuracies of 0.73 and less. This frame includes lexical units such as *can.v* and *able.a*, which are both frequent and unspecific and therefore somewhat difficult to learn without frame-specific tuning. The same is true for three other frames, POSSESSION, and TEMPORAL_COLLOCATION, and LOCATIVE_RELATION, which also have a high number of frequent, ambiguous predicates including modals and prepositions (*have.v, in.prep, on.prep*). The final frame, WEAPON, behaves rather differently in that the models perform almost equally well. Since the predicates in this frame form a coherent topic and tend to be low in ambiguity (*bomb.n, missile.n, shotgun.n*), they are quite easily learned with only generalized embeddings.

The analysis at the predicate level is shown in Table 3. We see a distinction very similar to the frame

Predicate	**BU+TD** **Prototype**	BU+TD Exemplar	BU Prototype	BU Exemplar
people.n	1.00	1.00	0.97	0.97
know.v	0.96	0.89	0.90	0.87
have.v	0.92	0.85	0.85	0.74
in.prep	0.91	0.69	0.80	0.59
can.v	0.91	0.59	0.29	0.62

Table 3: Accuracies for top 5 predicates from Bottom-up+Top-down Prototype model across four model

level between high-ambiguity and low-ambiguity predicates. Highly frequent, ambiguous predicates such as *have.v*, *know.v*, *can.v*, and *in.prep* profit hugely from frame-specific fine-tuning since their pre-trained, contextualized embeddings are presumably more widely spread out. In contrast, the *people.n* predicate performs well in all models including the bottom-up ones.

6 Discussion and Conclusions

In this paper, we have taken up an old strand of research in cognitive psychology, categorization, and demonstrated how such research contributes to computational lexical semantics. We have argued that theories of categorization have something valuable to offer to neural embedding-based models of natural language semantics, namely a framework in which to ground model design and understand their consequences. We have considered two dimensions: (a) the distinction between prototype and exemplar categorization, where prototype models produce a summary representation of its categories, while exemplar models represent the input objects themselves; and (b) the decision between pure similarity-driven "bottom-up" categorization, and task-specific "top-down" categorization, which finds its direct counterpart in current embedding models in the distinction between pre-trained and fine-tuned embeddings.

Along these two dimensions, we have defined four models for frame-semantic frame identification with BERT embeddings. Empirically, we found that for this task it worked best (a) to learn category representations via a prototype, and (b) to fine-tune the representations with a small amount of frame-labeled data. Further analysis showed that the benefit of the fine-tuning was in particular to improve model performance in the face of *abstractness* and *ambiguity*: while all models work well on frames describing coherent, concrete topics and containing concrete predicates drawn from their topics (WEAPONS), only fine-tuned models perform well on frames that capture abstract semantic generalizations that do not correspond to coherent regions in embedding space (LOCATIVE_RELATION) or ambiguous predicates such as the predicate *can.v*, which is able to evoke five frames: PRESERVING, CAPABILITY, LIKELIHOOD, and POSSIBILITY.

While the benefit of fine-tuning is expected based on previous work, the relative performance of prototype and exemplar models was less predictable. Our analysis indicates that the outcome of our study – a win for prototypes – is presumably tied to the studies' use of full-text frame annotation, which can be exploited straightforwardly in a prototype setting to tune the long-distance dependencies captured by BERT's self-attention mechanisms.

References

Baker, C. F., C. J. Fillmore, and J. B. Lowe (1998). The Berkeley FrameNet project. In *Proceedings of ACL/COLING*, Montreal, QC, pp. 86–90.

Bellet, A., A. Habrard, and M. Sebban (2013). A survey on metric learning for feature vectors and structured data. *CoRR abs/1306.6709*.

Daelemans, W. and A. van den Bosch (2005). *Memory-based Language Processing*. Studies in natural language processing. Cambridge University Press.

Das, D., D. Chen, A. F. Martins, N. Schneider, and N. A. Smith (2014). Frame-semantic parsing. *Computational Linguistics 40*(1), 9–56.

Das, D., N. Schneider, D. Chen, and N. A. Smith (2010). Probabilistic frame-semantic parsing. In *Proceedings of NAACL/HLT*, Los Angeles, California, pp. 948–956.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Erk, K. (2009). Representing words as regions in vector space. In *Proceedings of CoNLL-2009*, Boulder, CO, pp. 57–65.

Erk, K. and S. Padó (2010). Exemplar-based models for word meaning in context. In *Proceedings of ACL*, Uppsala, Sweden, pp. 92–97.

Fillmore, C. J. (1982). Frame Semantics. In *Linguistics in the Morning Calm*, pp. 111–138. Seoul, Korea: Hanshin.

Gardner, M., J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer (2018). Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pp. 1–6. Association for Computational Linguistics.

Gildea, D. and D. Jurafsky (2002). Automatic labeling of semantic roles. *Computational Linguistics 28*(3), 245–288.

Green, R., B. J. Dorr, and P. Resnik (2004). Inducing frame semantic verb classes from WordNet and LDOCE. In *Proceedings of ACL*, Barcelona, Spain, pp. 375–382.

Harris, Z. S. (1954). Distributional structure. *Word 10*(2-3), 146–162.

Hartmann, S., I. Kuznetsov, T. Martin, and I. Gurevych (2017). Out-of-domain framenet semantic role labeling. In *Proceedings of EACL*, Valencia, Spain, pp. 471–482.

Hermann, K. M., D. Das, J. Weston, and K. Ganchev (2014). Semantic frame identification with distributed word representations. In *Proceedings of ACL*, Baltimore, MD, pp. 1448–1458.

Hill, F., K. Cho, S. Jean, and Y. Bengio (2017). The representational geometry of word meanings acquired by neural machine translation models. *Machine Translation 31*(1–2), 3–18.

Hintzman, D. L. (1986). "Schema abstraction" in a multiple-trace memory model. *Psychological review 93*(4), 411.

Howard, J. and S. Ruder (2018). Universal language model fine-tuning for text classification. In *Proceedings of ACL*, Melbourne, Australia, pp. 328–339.

Lieto, A., D. P. Radicioni, and V. Rho (2017). Dual peccs: a cognitive system for conceptual representation and categorization. *Journal of Experimental & Theoretical Artificial Intelligence 29*(2), 433–452.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, Lake Tahoe, NV, pp. 3111–3119.

Mitchell, J. and M. Lapata (2008). Vector-based models of semantic composition. In *Proceedings of ACL*, Columbus, OH, pp. 236–244.

Murphy, G. L. (2002). *The Big Book of Concepts*. Boston, MA: MIT Press.

Nosofsky, R. M. (1986). Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General 115*(1), 39–61.

Nosofsky, R. M. and S. R. Zaki (2002). Exemplar and prototype models revisited: Response strategies, selective attention, and stimulus generalization. *Journal of Experimental Psychology: Learning, Memory, and Cognition 28*(5), 924–940.

Peng, H., S. Thomson, S. Swayamdipta, and N. A. Smith (2018). Learning joint semantic parsers from disjoint data. In *Proceedings of NAACL*, pp. 1492–1502.

Posner, M. I. and S. W. Keele (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology 77*(3), 353–363.

Radford, A., K. Narasimhan, T. Salimans, and I. Sutskever (2018). Improving language understanding with unsupervised learning. Technical report, OpenAI.

Reisinger, J. and R. J. Mooney (2010). Multi-prototype vector-space models of word meaning. In *Proceedings of HLT/NAACL*, Los Angeles, CA, pp. 109–117.

Rosch, E. (1975). Cognitive representations of semantic categories. *Journal of Experimental Psychology: General 104*, 192–233.

Smith, E. and S. A. Sloman (1996). Similarity- versus rule-based categorization. *Memory & Cognition 22*(4), 377–386.

Socher, R., A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. 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*, pp. 1631–1642. Association for Computational Linguistics.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Proceedings of NIPS*, Long Beach, CA, pp. 5998–6008.

Yang, B. and T. Mitchell (2017). A joint sequential and relational model for frame-semantic parsing. In *Proceedings of EMNLP*, Copenhagen, Denmark, pp. 1247–1256.

Zhu, Y., R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. 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*, Santiago, Chile, pp. 19–27.