Advanced techniques for through and contextually Interpreting Noun-Noun Compounds

Abstract

This study examines the effectiveness of transfer learning and multi-task learning in the context of a complex semantic classification problem: understanding the meaning of noun-noun compounds. Through a series of detailed experiments and an in-depth analysis of errors, we demonstrate that employing transfer learning by initializing parameters and multi-task learning through parameter sharing enables a neural classification model to better generalize across a dataset characterized by a highly uneven distribution of semantic relationships. Furthermore, we illustrate how utilizing dual annotations, which involve two distinct sets of relations applied to the same compounds, can enhance the overall precision of a neural classifier and improve its F1 scores for less common yet more challenging semantic relations.

1 Introduction

Noun-noun compound interpretation involves determining the semantic connection between two nouns (or noun phrases in multi-word compounds). For instance, in the compound "street protest," the task is to identify the semantic relationship between "street" and "protest," which is a locative relation in this example. Given the prevalence of noun-noun compounds in natural language and its significance to other natural language processing (NLP) tasks like question answering and information retrieval, understanding noun-noun compounds has been extensively studied in theoretical linguistics, psycholinguistics, and computational linguistics.

In computational linguistics, noun-noun compound interpretation is typically treated as an automatic classification task. Various machine learning (ML) algorithms and models, such as Maximum Entropy, Support Vector Machines, and Neural Networks, have been employed to decipher the semantics of nominal compounds. These models utilize information from lexical semantics, like WordNet-based features, and distributional semantics, such as word embeddings. However, noun-noun compound interpretation remains a challenging NLP problem due to the high productivity of noun-noun compounding as a linguistic structure and the difficulty in deriving the semantics of noun-noun compounds from their constituents. Our research contributes to advancing NLP research on noun-noun compound interpretation through the application of transfer and multi-task learning.

The application of transfer learning (TL) and multi-task learning (MTL) in NLP has gained significant attention in recent years, yielding varying outcomes based on the specific tasks, model architectures, and datasets involved. These varying results, combined with the fact that neither TL nor MTL has been previously applied to noun-noun compound interpretation, motivate our thorough empirical investigation into the use of TL and MTL for this task. Our aim is not only to add to the existing research on the effectiveness of TL and MTL for semantic NLP tasks generally but also to ascertain their specific advantages for compound interpretation.

A key reason for utilizing multi-task learning is to enhance generalization by making use of the domain-specific details present in the training data of related tasks. In this study, we demonstrate that TL and MTL can serve as a form of regularization, enabling the prediction of infrequent relations within a dataset marked by a highly skewed distribution of relations. This dataset is particularly well-suited for TL and MTL experimentation, as elaborated in Section 3.

Our contributions are summarized as follows:

1. Through meticulous analysis of results, we discover that TL and MTL, especially when applied to the embedding layer, enhance overall accuracy and F1 scores for less frequent relations in a highly skewed dataset, compared to a robust single-task learning baseline. 2. Although our research concentrates on TL and MTL, we present, to our knowledge, the first experimental results on the relatively recent dataset from Fares (2016).

2 Related Work

Approaches to interpreting noun-noun compounds differ based on the classification of compound relations, as well as the machine learning models and features employed to learn these relations. For instance, some define a broad set of relations, while others employ a more detailed classification. Some researchers challenge the idea that noun-noun compounds can be interpreted using a fixed, predetermined set of relations, proposing alternative methods based on paraphrasing. We center our attention on methods that frame the interpretation problem as a classification task involving a fixed, predetermined set of relations. Various machine learning models have been applied to this task, including nearest neighbor classifiers that use semantic similarity based on lexical resources, kernel-based methods like SVMs that utilize lexical and relational features, Maximum Entropy models that incorporate a wide range of lexical and surface form features, and neural networks that rely on word embeddings or combine word embeddings with path embeddings. Among these studies, some have utilized the same dataset. To our knowledge, TL and MTL have not been previously applied to compound interpretation. Therefore, we review prior research on TL and MTL in other NLP tasks.

Several recent studies have conducted extensive experiments on the application of TL and MTL to a variety of NLP tasks, such as named entity recognition, semantic labeling, sentence-level sentiment classification, super-tagging, chunking, and semantic dependency parsing. The consensus among these studies is that the advantages of TL and MTL are largely contingent on the characteristics of the tasks involved, including the unevenness of the data distribution, the semantic relatedness between the source and target tasks, the learning trajectory of the auxiliary and main tasks (where target tasks that quickly reach a plateau benefit most from non-plateauing auxiliary tasks), and the structural similarity between the tasks. Besides differing in the NLP tasks they investigate, the aforementioned studies employ slightly varied definitions of TL and MTL. Our research aligns with certain studies in that we apply TL and MTL to learn different semantic annotations of noun-noun compounds using the same dataset. However, our experimental design is more akin to other work in that we experiment with initializing parameters across all layers of the neural network and concurrently train a single MTL model on two sets of relations.

3 Task Definition and Dataset

The objective of this task is to train a model to categorize the semantic relationships between pairs of nouns in a labeled dataset, where each pair forms a noun-noun compound. The complexity of this task is influenced by factors such as the label set used and its distribution. For the experiments detailed in this paper, we utilize a noun-noun compounds dataset that features compounds annotated with two distinct taxonomies of relations. This means that each noun-noun compound is associated with two different relations, each based on different linguistic theories. This dataset is derived from established linguistic resources, including NomBank and the Prague Czech-English Dependency Treebank 2.0 (PCEDT). We chose this dataset for two primary reasons: firstly, the dual annotation of relations on the same set of compounds is ideal for exploring TL and MTL approaches; secondly, aligning two different annotation frameworks on the same data allows for a comparative analysis across these frameworks.

Specifically, we use a portion of the dataset, focusing on type-based instances of two-word compounds. The original dataset also encompasses multi-word compounds (those made up of more than two nouns) and multiple instances per compound type. We further divide the dataset into three parts: training, development, and test sets. Table 1 details the number of compound types and the vocabulary size for each set, including a breakdown of words appearing in the right-most (right constituents) and left-most (left constituents) positions. The two label sets consist of 35 PCEDT functors and 18

NomBank argument and adjunct relations. As discussed in Section 7.1, these label sets have a highly uneven distribution.

Table 1: Characteristics of the noun-noun compound dataset used in our experiments. The numbers in this table correspond to a subset of the dataset, see Section 3.

	Train	Dev	Test
Compounds	6932	920	1759
Vocab size	4102	1163	1772
Right constituents	2304	624	969
Left constituents	2405	618	985

Many relations in PCEDT and NomBank conceptually describe similar semantic ideas, as they are used to annotate the semantics of the same text. For instance, the temporal and locative relations in NomBank (ARGM-TMP and ARGM-LOC, respectively) and their PCEDT counterparts (TWHEN and LOC) exhibit relatively consistent behavior across frameworks, as they annotate many of the same compounds. However, some relations that are theoretically similar do not align well in practice. For example, the functor AIM in PCEDT and the modifier argument ARGM-PNC in NomBank express a somewhat related semantic concept (purpose), but there is minimal overlap between the sets of compounds they annotate. Nevertheless, it is reasonable to assume that the semantic similarity in the label sets, where it exists, can be leveraged through transfer and multi-task learning, especially since the overall distribution of relations differs between the two frameworks.

4 Transfer vs. Multi-Task Learning

In this section, we employ the terminology and definitions established by Pan and Yang (2010) to articulate our framework for transfer and multi-task learning. Our classification task can be described in terms of all training pairs (X, Y) and a probability distribution P(X), where X represents the input feature space, Y denotes the set of all labels, and N is the training data size. The domain of a task is defined by X, P(X). Our goal is to learn a function f(X) that predicts Y based on the input features X. Considering two ML tasks, Ta and Tb, we would train two distinct models to learn separate functions fa and fb for predicting Ya and Yb in a single-task learning scenario. However, if Ta and Tb are related, either explicitly or implicitly, TL and MTL can enhance the generalization of either or both tasks. Two tasks are deemed related when their domains are similar but their label sets differ, or when their domains are dissimilar but their label sets are identical. Consequently, noun-noun compound interpretation using the dataset is well-suited for TL and MTL, as the training examples are identical, but the label sets are distinct.

For clarity, we differentiate between transfer learning and multi-task learning in this paper, despite these terms sometimes being used interchangeably in the literature. We define TL as the utilization of parameters from a model trained on Ta to initialize another model for Tb. In contrast, MTL involves training parts of the same model to learn both Ta and Tb, essentially learning one set of parameters for both tasks. The concept is to train a single model simultaneously on both tasks, where one task introduces an inductive bias that aids the model in generalizing over the main task. It is important to note that this does not necessarily imply that we aim to use a single model to predict both label sets in practice.

5 Neural Classification Models

This section introduces the neural classification models utilized in our experiments. To discern the impact of TL and MTL, we initially present a single-task learning model, which acts as our baseline. Subsequently, we employ this same model to implement TL and MTL.

5.1 Single-Task Learning Model

In our single-task learning (STL) configuration, we train and fine-tune a feed-forward neural network inspired by the neural classifier proposed by Dima and Hinrichs (2015). This network comprises four layers: 1) an input layer, 2) an embedding layer, 3) a hidden layer, and 4) an output layer. The input

layer consists of two integers that indicate the indices of a compound's constituents in the embedding layer, where the word embedding vectors are stored. These selected vectors are then passed to a fully connected hidden layer, the size of which matches the dimensionality of the word embedding vectors. Finally, a softmax function is applied to the output layer to select the most probable relation.

The compound's constituents are represented using a 300-dimensional word embedding model trained on an English Wikipedia dump and the English Gigaword Fifth Edition. The embedding model was trained by Fares et al. (2017). If a word is not found during lookup in the embedding model, we check if the word is uppercased and attempt to find the lowercase version. For hyphenated words not found in the embedding vocabulary, we split the word at the hyphen and average the vectors of its parts, if they are present in the vocabulary. If the word remains unrepresented after these steps, a designated vector for unknown words is employed.

5.1.1 Architecture and Hyperparameters

Our selection of hyperparameters is informed by multiple rounds of experimentation with the single-task learning model, as well as the choices made by prior work. The weights of the embedding layer are updated during the training of all models. We utilize the Adaptive Moment Estimation (Adam) optimization function across all models, with a learning rate set to 0.001. The loss function employed is the negative-log likelihood. A Sigmoid activation function is used for the units in the hidden layer. All models are trained with mini-batches of size five. The maximum number of epochs is capped at 50, but an early stopping criterion based on the model's accuracy on the validation split is also implemented. This means that training is halted if the validation accuracy does not improve over five consecutive epochs. All models are implemented in Keras, using TensorFlow as the backend. The TL and MTL models are trained using the same hyperparameters as the STL model.

5.2 Transfer Learning Models

In our experiments, transfer learning involves training an STL model on PCEDT relations and then using some of its weights to initialize another model for NomBank relations. Given the neural classifier architecture detailed in Section 5.1, we identify three ways to implement TL: 1) TLE: Transferring the embedding layer weights, 2) TLH: Transferring the hidden layer weights, and 3) TLEH: Transferring both the embedding and hidden layer weights. Furthermore, we differentiate between transfer learning from PCEDT to NomBank and vice versa. This results in six setups, as shown in Table 2. We do not apply TL (or MTL) to the output layer because it is task- or dataset-specific.

5.3 Multi-Task Learning Models

In MTL, we train a single model to simultaneously learn both PCEDT and NomBank relations, meaning all MTL models have two objective functions and two output layers. We implement two MTL setups: MTLE, which features a shared embedding layer but two task-specific hidden layers, and MTLF, which has no task-specific layers aside from the output layer (i.e., both the embedding and hidden layers are shared). We distinguish between the auxiliary and main tasks based on which validation accuracy (NomBank's or PCEDT's) is monitored by the early stopping criterion. This leads to a total of four MTL models, as shown in Table 3.

6 Experimental Results

Tables 2 and 3 display the accuracies of the various TL and MTL models on the development and test splits for NomBank and PCEDT. The top row in both tables indicates the accuracy of the STL model. All models were trained solely on the training split. Several insights can be gleaned from these tables. Firstly, the accuracy of the STL models decreases when evaluated on the test split for both NomBank and PCEDT. Secondly, all TL models achieve improved accuracy on the NomBank test split, although transfer learning does not significantly enhance accuracy on the development split of the same dataset. The MTL models, especially MTLF, have a detrimental effect on the development accuracy of NomBank, yet we observe a similar improvement, as with TL, on the test split. Thirdly, both TL and MTL models demonstrate less consistent effects on PCEDT (on both development and test splits) compared to NomBank. For instance, all TL models yield an absolute improvement of

about 1.25 points in accuracy on NomBank, whereas in PCEDT, TLE clearly outperforms the other two TL models (TLE improves over the STL accuracy by 1.37 points).

Table 2: Accuracy (%) of the transfer learning models.

Model	NomBank		PCEDT	
	Dev	Test	Dev	Test
STL	78.15	76.75	58.80	56.05
TLE	78.37	78.05	59.57	57.42
TLH	78.15	78.00	59.24	56.51
TLEH	78.48	78.00	59.89	56.68

Table 3: Accuracy (%) of the MTL models.

Model	NomBank		PCEDT	
	Dev	Test	Dev	Test
STL	78.15	76.75	58.80	56.05
MTLE	77.93	78.45	59.89	56.96
MTLF	76.74	78.51	58.91	56.00

Overall, the STL models' accuracy declines when tested on the NomBank and PCEDT test splits, compared to their performance on the development split. This could suggest overfitting, especially since our stopping criterion selects the model with the best performance on the development split. Conversely, TL and MTL enhance accuracy on the test splits, despite using the same stopping criterion as STL. We interpret this as an improvement in the models' ability to generalize. However, since these improvements are relatively minor, we further analyze the results to understand if and how TL and MTL are beneficial.

7 Results Analysis

This section provides a detailed analysis of the models' performance, drawing on insights from the dataset and the classification errors made by the models. The discussion in the following sections is primarily based on the results from the test split, as it is larger than the development split.

7.1 Relation Distribution

To illustrate the complexity of the task, we depict the distribution of the most frequent relations in NomBank and PCEDT across the three data splits in Figure 1. Notably, approximately 71.18% of the relations in the NomBank training split are of type ARG1 (prototypical patient), while 52.20% of the PCEDT relations are of type RSTR (an underspecified adnominal modifier). Such a highly skewed distribution makes learning some of the other relations more challenging, if not impossible in certain cases. In fact, out of the 15 NomBank relations observed in the test split, five are never predicted by any of the STL, TL, or MTL models. Similarly, of the 26 PCEDT relations in the test split, only six are predicted. However, the unpredicted relations are extremely rare in the training split (e.g., 23 PCEDT functors appear less than 20 times), making it doubtful whether any ML model could learn them under any circumstances.

Given this imbalanced distribution, it is evident that accuracy alone is insufficient to determine the best-performing model. Therefore, in the subsequent section, we report and analyze the F1 scores of the predicted NomBank and PCEDT relations across all STL, TL, and MTL models.

7.2 Per-Relation F1 Scores

Tables 4 and 5 present the per-relation F1 scores for NomBank and PCEDT, respectively. We only include results for relations that are actually predicted by at least one of the models.

Table 4: Per-label F1 score on the NomBank test split.

	A0	A1	A2	A3	LOC	MNR	TMP
Count	132	1282	153	75	25	25	27
STL	49.82	87.54	45.78	60.81	28.57	29.41	66.67
TLE	55.02	87.98	41.61	60.14	27.91	33.33	63.83
TLH	54.81	87.93	42.51	60.00	25.00	35.29	65.31
TLEH	53.62	87.95	42.70	61.11	29.27	33.33	65.22
MTLE	54.07	88.34	42.86	61.97	30.00	28.57	66.67
MTLF	53.09	88.41	38.14	62.69	00.00	00.00	52.17

Table 5: Per-label F1 score on the PCEDT test split.

	ACT	TWHEN	APP	PAT	REG	RSTR
Count	89	14	118	326	216	900
STL	43.90	42.11	22.78	42.83	20.51	68.81
TLE	49.37	70.97	27.67	41.60	30.77	69.67
TLH	53.99	62.07	25.00	43.01	26.09	68.99
TLEH	49.08	64.52	28.57	42.91	28.57	69.08
MTLE	54.09	66.67	24.05	42.03	27.21	69.31
MTLF	47.80	42.11	25.64	40.73	19.22	68.89

Several noteworthy patterns emerge from Tables 4 and 5. Firstly, the MTLF model appears to be detrimental to both datasets, leading to significantly degraded F1 scores for four NomBank relations, including the locative modifier ARGM-LOC and the manner modifier ARGM-MNR (abbreviated as LOC and MNR in Table 4), which the model fails to predict altogether. This same model exhibits the lowest F1 score compared to all other models for two PCEDT relations: REG (expressing a circumstance) and PAT (patient). Considering that the MTLF model achieves the highest accuracy on the NomBank test split (as shown in Table 3), it becomes even more apparent that relying solely on accuracy scores is inadequate for evaluating the effectiveness of TL and MTL for this task and dataset.

Secondly, with the exception of the MTLF model, all TL and MTL models consistently improve the F1 score for all PCEDT relations except PAT. Notably, the F1 scores for the relations TWHEN and ACT show a substantial increase compared to other PCEDT relations when only the embedding layer's weights are shared (MTLE) or transferred (TLE). This outcome can be partially understood by examining the correspondence matrices between NomBank arguments and PCEDT functors, presented in Tables 7 and 6. These tables illustrate how PCEDT functors map to NomBank arguments in the training split (Table 6) and vice versa (Table 7). Table 6 reveals that 80% of the compounds annotated as TWHEN in PCEDT were annotated as ARGM-TMP in NomBank. Additionally, 47% of ACT (Actor) relations map to ARG0 (Proto-Agent) in NomBank. While this mapping is not as distinct as one might hope, it is still relatively high when compared to how other PCEDT relations map to ARG0. The correspondence matrices also demonstrate that the presumed theoretical similarities between NomBank and PCEDT relations do not always hold in practice. Nevertheless, even such imperfect correspondences can provide a training signal that assists the TL and MTL models in learning relations like TWHEN and ACT.

Since the TLE model outperforms STL in predicting REG by ten absolute points, we examined all REG compounds correctly classified by TLE but misclassified by STL. We found that STL misclassified them as RSTR, indicating that TL from NomBank helps TLE recover from STL's overgeneralization in RSTR prediction.

The two NomBank relations that receive the highest boost in F1 score (about five absolute points) are ARG0 and ARGM-MNR, but the improvement in the latter corresponds to only one additional compound, which might be a chance occurrence. Overall, TL and MTL from NomBank to PCEDT are more helpful than the reverse. One explanation is that five PCEDT relations (including the four most frequent ones) map to ARG1 in NomBank in more than 60% of cases for each relation, as seen in the first rows of Tables 6 and 7. This suggests that the weights learned to predict PCEDT relations

Table 6: Correspondence matrix between PCEDT functors and NomBank arguments. Slots with '-' indicate zero, 0.00 represents a very small number but not zero.

	A1	A2	A0	A3	LOC	TMP	MNR
RSTR	0.70	0.11	0.06	0.06	0.02	0.01	0.02
PAT	0.90	0.05	0.01	0.02	0.01	-	0.00
REG	0.78	0.10	0.04	0.06	0.00	0.00	0.00
APP	0.62	0.21	0.13	0.02	0.01	0.00	-
ACT	0.47	0.03	0.47	0.01	0.01	-	0.01
AIM	0.65	0.12	0.07	0.06	0.01	-	-
TWHEN	0.10	0.03	-	-	-	0.80	-
Count	3617	1312	777	499	273	116	59

Table 7: Correspondence matrix between NomBank arguments and PCEDT functors.

	RSTR	PAT	REG	APP	ACT	AIM	TWHEN
A1	0.51	0.54	0.12	0.06	0.03	0.02	0.00
A2	0.47	0.09	0.11	0.14	0.01	0.02	0.00
A0	0.63	0.03	0.07	0.13	0.26	0.02	-
A3	0.66	0.08	0.13	0.03	0.01	0.02	-
LOC	0.36	0.07	0.02	0.05	0.03	0.01	-
TMP	0.78	-	0.01	0.01	-	-	0.01
MNR	0.24	0.05	0.01	-	0.03	-	-
Count	4932	715	495	358	119	103	79

offer little to no inductive bias for NomBank relations. Conversely, the mapping from NomBank to PCEDT shows that although many NomBank arguments map to RSTR in PCEDT, the percentages are lower, making the mapping more diverse and discriminative, which seems to aid TL and MTL models in learning less frequent PCEDT relations.

To understand why the PCEDT functor AIM is never predicted despite being more frequent than TWHEN, we found that AIM is almost always misclassified as RSTR by all models. Furthermore, AIM and RSTR have the highest lexical overlap in the training set among all PCEDT relation pairs: 78.35% of left constituents and 73.26% of right constituents of compounds annotated as AIM occur in other compounds annotated as RSTR. This explains the models' inability to learn AIM but raises questions about their ability to learn relational representations, which we explore further in Section 7.3.

Table 8: Macro-average F1 score on the test split.

Model	NomBank	PCEDT
STL	52.66	40.15
TLE	52.83	48.34
TLH	52.98	46.52
TLEH	53.31	47.12
MTLE	53.21	47.23
MTLF	42.07	40.73

Finally, to demonstrate the benefits of TL and MTL for NomBank and PCEDT, we report the F1 macro-average scores in Table 8. This is arguably the appropriate evaluation measure for imbalanced classification problems. Note that relations not predicted by any model are excluded from the macro-average calculation. Table 8 clearly shows that TL and MTL on the embedding layer yield significant improvements for PCEDT, with about a 7-8 point increase in macro-average F1, compared to just 0.65 in the best case for NomBank.

7.3 Generalization on Unseen Compounds

We now analyze the models' ability to generalize to compounds not seen during training. Recent research suggests that gains in noun-noun compound interpretation using word embeddings and similar neural classification models might be due to lexical memorization. In other words, the models learn that specific nouns are strong indicators of specific relations. To assess the role of lexical memorization in our models, we quantify the number of unseen compounds that the STL, TL, and MTL models predict correctly.

We differentiate between 'partly' and 'completely' unseen compounds. A compound is 'partly' unseen if one of its constituents (left or right) is not present in the training data. A 'completely' unseen compound is one where neither the left nor the right constituent appears in the training data. Overall, nearly 20% of the compounds in the test split have an unseen left constituent, about 16% have an unseen right constituent, and 4% are completely unseen. Table 9 compares the performance of the different models on these three groups in terms of the proportion of compounds misclassified in each group.

Table 9: Generalization error on the subset of unseen compounds in the test split. L: Left constituent. R: Right constituent. L&R: Completely unseen.

	NomBank			PCEDT		
Model	L	R	L&R	L	R	L&R
Count	351	286	72	351	286	72
STL	27.92	39.51	50.00	45.01	47.55	41.67
TLE	25.93	36.71	48.61	43.87	47.55	41.67
TLH	26.21	38.11	50.00	46.15	49.30	47.22
TLEH	26.50	38.81	52.78	45.87	47.55	43.06
MTLE	24.50	33.22	38.89	44.44	47.20	43.06
MTLF	22.79	34.27	40.28	44.16	47.90	38.89

Table 9 shows that Transfer Learning (TL) and Multi-Task Learning (MTL) approaches reduce generalization error in NomBank across all scenarios, with the exception of TLH and TLEH for completely unseen compounds, where error increases. The greatest error reductions are achieved by MTL models across all three types of unseen compounds. Specifically, MTLE reduces the error by approximately six points for compounds with unseen right constituents and by eleven points for fully unseen compounds. Moreover, MTLF reduces the error by five points when the left constituent is unseen. It's important to interpret these results in conjunction with the Count row in Table 9 for a comprehensive view. For example, the eleven-point error decrease in fully unseen compounds represents eight compounds. In PCEDT, the largest error reduction is on unseen left constituents, which is about 1.14 points, corresponding to four compounds; it's 0.35 on unseen right constituents (one compound) and 2.7 on fully unseen compounds, or two compounds.

Upon manual inspection of compounds that led to substantial reductions in the generalization error, specifically within NomBank, we examined the distribution of relations within correctly predicted unseen compound sets. Compared to the STL model, MTLE reduces generalization error for completely unseen compounds by a total of eight compounds, of which seven are annotated with the relation ARG1, which is the most common in NomBank. Regarding the unseen right constituents, MTLE's 24 improved compounds consist of 18 ARG1, 5 ARG0, and 1 ARG2 compounds. A similar pattern arises when examining TLE model improvements, where most gains come from better predictions of ARG1 and ARG0 relations.

A large portion of unseen compounds, whether partly or entirely unseen, that were misclassified by every model, were not of type ARG1 in NomBank, or RSTR in PCEDT. This pattern, along with correctly predicted unseen compounds primarily annotated with the most common relations, suggests that classification models rely on lexical memorization to learn the compound relation interpretation.

To better comprehend lexical memorization's impact, we present the ratio of relation-specific constituents in both NomBank and PCEDT, as depicted in Figure 2. We define a relation-specific constituent as a left or right constituent that appears with only one specific relation within the training data. Its ratio is calculated as its proportion in the full set of left or right constituents for each

relation. Analyzing Figure 2 reveals that NomBank relations possess higher ratios of relation-specific constituents compared to PCEDT. This potentially makes learning the former easier if the model solely relies on lexical memorization. Additionally, ARGM-TMP in NomBank and TWHEN in PCEDT have distinctly high ratios compared to other relations in Figure 2. These relations also have the second-highest F1 score in their datasets—except for STL on PCEDT (see Tables 4 and 5). Lexical memorization is therefore a likely cause of these high F1 scores. We also observed that lower ratios of relation-specific constituents correlate with lower F1 scores, such as APP and REG in PCEDT. Based on these insights, we can't dismiss the possibility that our models show some degree of lexical memorization, despite manual analysis also presenting cases where models demonstrate generalization and correct predictions in situations where lexical memorization is impossible.

8 Conclusion

The application of transfer and multi-task learning in natural language processing has gained significant traction, yet considerable ambiguity persists regarding the effectiveness of particular task characteristics and experimental setups. This research endeavors to clarify the benefits of TL and MTL in the context of semantic interpretation of noun-noun compounds. By executing a sequence of minimally contrasting experiments and conducting thorough analysis of results and prediction errors, we demonstrate how both TL and MTL can mitigate the effects of class imbalance and drastically enhance predictions for low-frequency relations. Overall, our TL, and particularly our MTL models, are better at making predictions both quantitatively and qualitatively. Notably, the improvements are observed on the 'most challenging' inputs that include at least one constituent that was not present in the training data. However, clear indications of 'lexical memorization' effects are evident in our error analysis of unseen compounds.

Typically, the transfer of representations or sharing between tasks is more effective at the embedding layers, which represent the model's internal representation of the compound constituents. Furthermore, in multi-task learning, the complete sharing of model architecture across tasks degrades its capacity to generalize when it comes to less frequent relations.

The dataset provided by Fares (2016) is an appealing resource for new neural approaches to compound interpretation because it links this sub-problem with broad-coverage semantic role labeling or semantic dependency parsing in PCEDT and NomBank. Future research will focus on incorporating additional natural language processing tasks defined using these frameworks to understand noun-noun compound interpretation using TL and MTL.