# Hydra: A Multi-headed Approach to Frame-Semantic Parsing

**Anonymous ACL submission**

## Abstract

We present Hydra, a transformer-based system that achieves state-of-the-art performance in identifying targets and frame elements in the FrameNet dataset. Our target identification model, Hydra-T, utilizes a modified prefix tree to support multi-word lexical units more effectively, allowing for the identification of split lexical units and achieving a +0.16 improvement in F1 score over the previous state-of-the-art model. Our frame identification model, Hydra-F, improves upon previous multiple-choice approaches by incorporating additional negative training samples, resulting in improved performance, particularly for frames with limited annotated samples, achieving an accuracy of 0.923 and a macro F1 score of 0.793. Our argument identification model, Hydra-A, presents a novel approach of defining a classification head for each frame to identify frame elements as a multi-task classification problem, resulting in an accuracy of 0.845 and a macro F1 score of 0.713. Unlike previous works, we also take into account the skewed distribution of annotated frames and frame elements in FrameNet by using macro-averaged performance metrics, rather than relying solely on accuracy, to evaluate our system's performance.

## 1 Introduction

Frame-semantic parsing (Gildea and Jurafsky, 2002) automatically identifies semantic frames and frame elements within text. Semantic frames (Fillmore et al., 2006) are events or situations which have frame elements that describe different roles within them. Semantic frames provide a structured framework for performing and explaining natural language processing tasks such as knowledge extraction (Søgaard et al., 2015), question answering (Gildea and Jurafsky, 2002), fact checking (Arslan et al., 2020), and event detection (Spiliopoulou et al., 2017). The frame-semantic annotations of an example sentence are shown in Figure 1.

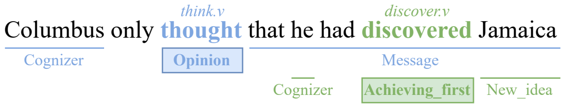

Figure 1: An example of frame-semantic annotations in FrameNet (Baker et al., 1998). Targets are represented by bold words with their corresponding lexical unit italicized above and evoked frame boxed below. Frame elements are labeled below the text and are color-coded to match their corresponding frame.

Frame-semantic parsing consists of three primary tasks: *target identification*, *frame identification*, and *argument identification* (Das et al., 2014). Target identification is spotting of targets—words or phrases that evoke frames in a sentence, e.g., *thought* and *discovered* in Figure 1. Each target is associated with a particular lexical unit, usually taking the form of *root.POS*, e.g., *think.v* for *thought* and *discover.v* for *discovered* in Figure 1. Frame identification is the classification of each target into its respective frame, e.g., OPINION and ACHIEVING_FIRST in Figure 1. Argument identification is the identification of each frame element belonging to a frame evoked in a sentence, e.g., "Columbus", the Cognizer frame element, and "that he had discovered Jamaica", the Message frame element, in the OPINION frame in Figure 1.

Models which perform each of the frame-semantic parsing tasks on an input sentence are end-to-end models (Swayamdipta et al., 2017), while models for a single task are referred to by their task, e.g., frame identification models. Building an end-to-end system typically requires a separate stage for each individual task (Swayamdipta et al., 2017; Das et al., 2010) since detecting frame elements requires knowledge of the evoked frame, and determining which frames are evoked typically requires knowledge of the targets.

In this paper, we present Hydra[1], a new frame-

[1] Our codebase and data can be found at `https://github.com/ofiuawq3h40of97wu3h4if/hydra`.

semantic parsing system which outperforms other systems in target identification and argument identification, with similar performance in frame identification. Its target identification model, Hydra-T, generates candidate targets for each lexical unit in the FrameNet 1.7 fulltext annotations (Baker et al., 1998) with a recall of 0.995. Previous works have approached target identification using a candidate generation and filtering process due to the poor performance of statistical models on directly predicting targets. This candidate generation process usually produces many false positives; we address this problem using a BERT-based transformer model (Devlin et al., 2019) to filter out the false-positives, a deviation from past works which used linguistic features to filter them out. This method results in a model which classifies tokens in the FrameNet 1.7 test set with an F1 score of 0.95, an improvement of +0.16 over SEMAFOR (Das et al., 2014), the previous state-of-the-art solution.

The system's frame identification model, Hydra-F, utilizes lexical unit definitions in a manner similar to the approach by FIDO (Jiang and Riloff, 2021). Hydra-F differs primarily in the way it learns to match a target with its corresponding frame and lexical unit definitions, particularly by randomly sampling negative frames in addition to the candidate frames generated from a lexical unit. This becomes especially important when the set of candidate frames is very small, or in most cases, is only a single frame. This difference in model training results in an accuracy of 0.923, an improvement of +0.014 over FIDO's implementation using lexical unit definitions performance on par with their full model. We also measure performance on par with KGFI (Su et al., 2021), the current state-of-the-art which has an accuracy of 0.924[2].

Our argument identification model, Hydra-A, defines separate classification heads for each frame, allowing the model to bias its weights independently of other frames, leading to each classification head focusing primarily on the frame elements relevant to its particular frame. To the best of our knowledge, this approach has not been previously explored in frame-semantic parsing. As an additional benefit, this allows our model to retain its weights for already-optimized classification heads, removing the need to retrain the entire model each time FrameNet is updated or new frames are added. With our approach, we reach a macro F1 score of 0.693, an increase of 0.152 over Open-SESAME (Swayamdipta et al., 2017), the previous state-of-the-art model for argument identification, while our micro and weighted F1 scores remain on par with theirs.

The paper's contributions are as follows.

- We propose a novel target identification model which generates candidate targets with an F1 score of 0.95 on the FrameNet 1.7 fulltext annotations, which is an improvement of 0.16 over the previous state-of-the-art.
- We propose an improvement on a standard frame identification training paradigm by forcing the model to differentiate between additional negative samples, allowing performance improvements on frames with few training samples.
- We propose a novel argument identification model which outperforms all previous works by introducing a classification head for each frame, approaching the problem as a multi-task classification problem.
- We perform novel experiments comparing different token aggregation methods which appear when using BERT or other word-piece-based tokenizers on frame-semantic parsing tasks.
- We propose and argue for the use of macro-averaged performance metrics in frame-semantic parsing solutions due to the heavy skew of frames and frame elements in FrameNet.

## 2 Background

FrameNet (Baker et al., 1998) is a collection of over 1,200 semantic frames, their respective inter-frame relationships, and thousands of annotated sentences spanning the frames. Each frame describes a distinct semantic idea and can be evoked by different words or phrases. For example, *discover*, *find*, and *invent* are all words which can evoke the ACHIEVING_FIRST frame. For each frame there are a varying number of corresponding frame elements, each of which indicates a role within the frame.

Previous works on frame-semantic parsing usually include one or more of the following: a target identification model, a frame identification model, or an argument identification model. The majority of previous works were on frame identification, with less being done to improve the performance of argument identification and target identification.

The most prominent and best performing target identification model is SEMAFOR (Das et al.,

[2] KGFI's performance cannot be reproduced since they have not released the source code of their model.

2014). SEMAFOR generates a master list of all targets and their morphological variants. It searches a given sentence for substrings from the master list to determine candidate targets. Then, a series of rules are applied to filter out candidate targets which do not actually occur in the sentence.

KGFI (Su et al., 2021) and FIDO (Jiang and Riloff, 2021) are two recent approaches to frame identification, both of which use pre-trained BERT models and attain similar performance. Prior to these works, the best-performing frame identification models were from Open-SESAME (Swayamdipta et al., 2017) and Peng et al., 2018.

KGFI proposes a frame identification solution which uses two graph convolutional networks (Kipf and Welling, 2017) to compute frame embeddings using contextualized BERT embeddings of each frame's definition and frame element relationship. FIDO creates a multiple-choice task, choosing the lexical unit-frame pair which best fits the input by incorporating the frame and lexical unit definitions into a BERT transformer model.

The current state-of-the-art model for end-to-end frame-semantic parsing and argument identification is Open-SESAME, which outperformed the previous-best performing system SEMAFOR. Open-SESAME uses GloVe (Pennington et al., 2014) word embeddings and a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) which classifies segments of text using a "syntactic scaffold" that learns features from other tasks, such as constituency parsing and part-of-speech tagging.

## 3 Target Identification

Target identification is the task of determining which tokens in a given sentence evoke a frame, such as *thought* and *discovered* in Figure 1. The primary distinction between Hydra-T and SEMAFOR is in the direction we approach the problem from. SEMAFOR generates a list of all morphological variants of targets in FrameNet and searches for them in a given sentence. This approach limits the ability to capture discontinuous targets, such as *there [would have] been* which corresponds to the lexical unit *there be.v*, especially when lexical units are not discontinuous due to verb forms, e.g., *take [the jacket] off* which corresponds to the *take off.v* lexical unit.

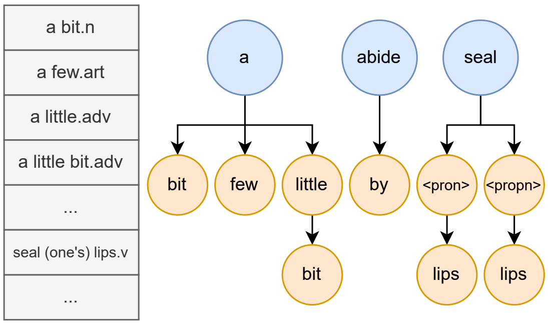

Figure 2: An example of the lexical unit prefix trees generated from some of the lexical units (grey) defined in FrameNet. Root nodes are colored in blue and child nodes are colored in orange.

### 3.1 Candidate Generation

We approach this task from the opposite direction, i.e., rather than searching an input sentence for substrings which occur in a master list of targets, we search for lexical units which are evoked in a sentence by using a set of prefix trees. We construct our prefix trees using lexical units defined in FrameNet as nodes indexed by the first word of its respective lexical unit. In the simplest and most common case, a single-word lexical unit exists as a prefix tree with a single node; however, many lexical units consist of multiple words in sequence, such as *a few.art* or *a little bit.adv*, oftentimes with auxiliary words among them, such as *seal (one's) lips.v*. To handle multi-word lexical units, each subsequent word is a child node of the previous word in the prefix trees, as shown in Figure 2. To further handle auxiliary words in multi-word lexical units, the prefix trees can also be searched using part-of-speech tags relevant to a particular lexical unit, e.g., <pron>/<propn> for *one's* or *someone's*, as also shown in Figure 2.

Given a sentence, to search the prefix trees, we traverse through each word in the sentence, attempting to match it to a node in a tree, starting from the root. To match a word to a node, we check if the node is indexed by any form of the word. After a word $w$ is matched with a node, we check if its subsequent word in the sentence, any of its word forms, or their parts-of-speech can match a child of the node. Additionally, we use a dependency parser to extract the dependents of $w$ (which may not come immediately after $w$) in the sentence and check if any of them exists as a child node.

To test the coverage of our candidate generation algorithm, we extract each sentence with at least one annotated target and their corresponding targets from FrameNet's fulltext annotations, resulting in $28,113$ targets from $5,071$ sentences. We apply our candidate generation algorithm to each sentence and identify target-frame pairs that are correctly detected by our algorithm. The importance of searching for target-frame pairs is that we want to ensure that our candidates spans are meaningful, otherwise, a method which produces all possible spans for a sentence would have perfect recall. Our algorithm is able to generate 27,971 of the targets in the fulltext annotations, giving it a recall of 0.995, and an additional 289,613 false positive candidate targets which need to be filtered out. The filtering is discussed in the next section.

### 3.2 Target Identification Model

Our target identification model, Hydra-T, uses a pretrained BERT model to filter out potential false positives—candidate targets that are not valid in the input sentence. Given a token sequence $x = (x_1, ..., x_n)$, Hydra-T begins by generating candidate targets $T^c = \{(i_1, j_1), ..., (i_m, j_m)\}$ (Section 3.1), where $m$ is the number of candidate targets generated, and $i$ and $j$ ($1 \leq i < j \leq n$) are the beginning and end of each candidate's span. Then, Hydra-T computes the contextualized BERT embeddings of each token $x^{emb} = (x_1^{emb}, ..., x_n^{emb})$, and use a binary classification layer to predict whether each span of tokens $X_{i:j}^{emb}$ is a valid target. To use these predicted spans in the frame identification model, postprocessing may be necessary to remove tokens in the target which are unrelated to the lexical unit. The weights of our binary classification layer are learned by minimizing the binary cross-entropy loss:

$$L = -\frac{1}{|T^c|} \sum_{i=1}^{|T^c|} y_i \log(x_i) + (1 - y_i) \log(1 - x_i) \tag{1}$$

where $y_i$ indicates whether span $i$ is a valid target and $x_i$ is the predicted probability.

## 4 Frame Identification

In frame identification, the input consists of a sequence of tokens $x = (x_1, x_2, ..., x_n)$ and a target $x_t \subset x$. The goal of this task is to determine which frame $f$ is evoked by $x_t$. In most cases, a target is a single word, but it could also be multiple consecutive, or even nonconsecutive, words.

It is important to note that sentences can, and in most cases do, evoke multiple frames, but each frame instance is evoked by a unique target in a sentence. Additionally, it is possible to have multiple occurrences of the same frame in a single sentence. For example, in "Bob *forgot* his keys at home yesterday, and today he *left* his wallet." the ABANDONMENT frame is evoked twice, by targets *forgot* and *left*.

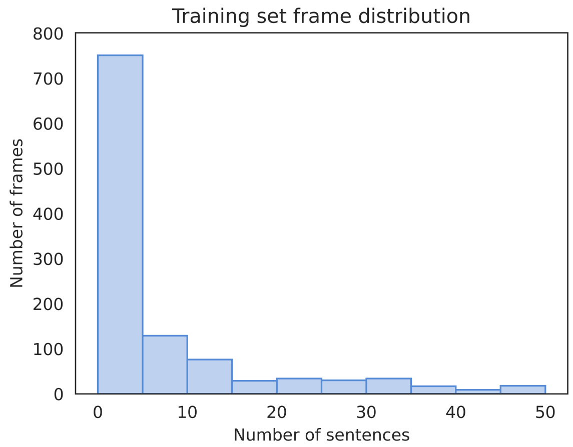

Figure 3: The distribution of frames in the FrameNet 1.7 training set. The right-most column in the histogram represents frames with 45 or more annotated sentences.

### 4.1 Lexicon Filtering

There are over 1,200 frames in FrameNet. Given a sentence and an identified target in the sentence, in theory each frame could be a possible candidate in frame identification. An inference model with that many possible choices would be daunting. In reality, though, most frames by FrameNet definition are evoked by a small set of lexical units. Likewise, for any given word, only a few frames can be evoked. For example, *thought* or its root word *think* can only evoke the frames AWARENESS, COGITATION, OPINION, and REGARD. This observation immediately leads to a simple idea called *lexicon filtering* for improving model performance (Das et al., 2014; Su et al., 2021). Specifically, using lexicon filtering, the number of candidate frames for the target *thought* is reduced from roughly 1,200 down to only 4, based on FrameNet definitions. On average, this results in 2.12 candidate frames per lexical unit in the FrameNet 1.7 fulltext annotations, and 1.29 candidate frames over all defined lexical units.

### 4.2 Frame and Lexical Unit Definitions

Recent work has shown that incorporating the definitions of frames and lexical units provided in the FrameNet dataset into the frame identification model results in improved performance (Jiang and Riloff, 2021; Su et al., 2021). This is particularly useful for frames with very few samples in the fulltext annotations, which are the overwhelming majority of frames, as seen in Figure 3.

We expand on the approach use by FIDO for representing the frame identification task as a multiple-choice classification task. The primary difference is in our problem formulation; FIDO uses frame and lexical unit definitions to assist in classifying the set of frames produced as a result of applying lexicon filtering. We find this approach to be insufficient as there is no information gained when lexicon filtering results in a single frame due to the class probability being 1 for that frame. In addition to the frames produced from lexicon filtering, we also randomly sample up to $N$ lexical units, where $N$ is the total number lexical units our model will choose from. This approach allows our model to generate additional negative samples for frames which have very few annotated samples.

### 4.3 Model

Given a tokenized input sentence $x$ and a target $x_t \subset x$, our frame identification model, Hydra-F, first applies lexicon filtering to the target $x_t$ to generate a set of candidate frames $F^c = (F^c_1, ..., F^c_N)$. For each candidate frame $f \in F^c$, we concatenate the definition of its lexical unit in $x_t$ to the target separated by a [SEP] token. Additionally, we randomly sample the remaining $N - |F^c|$ negative lexical unit definitions to the target, resulting in $N$ target-lexical unit definition pairs. For each additional negative lexical unit, we follow the same process. This results in $N$ total frames for our model to choose the best target-frame pair from.

Next, the model computes contextualized embeddings $x^{emb}$ from the concatenated input tokens using a pre-trained BERT model and computes an aggregate target embedding using an aggregation function $G$ on the embeddings corresponding to the tokens in the target. Finally, the model computes a classification score $s$ for each frame $f \in F^c$ using the following equation:

$$S(t, f) = G(x_t^{emb}) \cdot W_f \qquad (2)$$

where $W_f$ are the weights learned by minimizing the cross-entropy loss:

$$L = -\log \frac{\exp(S(t, f))}{\sum_{i=1}^{N} \exp(S(t, F_i))} \qquad (3)$$

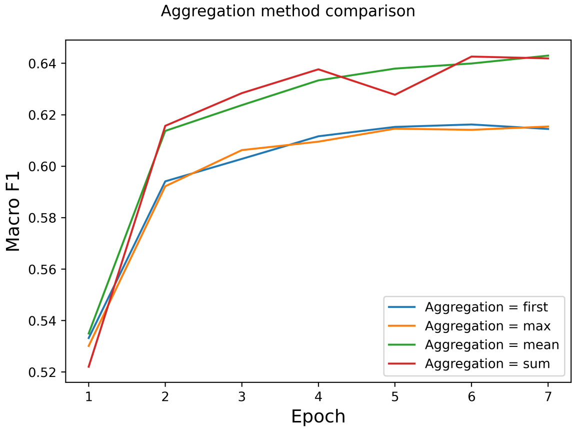

Figure 4: Macro F1 score of models used in aggregation method test. Here we can see a clear separation of $G_{mean}$ and $G_{sum}$ from $G_{first}$ and $G_{max}$.

We have tested four different aggregation functions: $G_{mean}$, $G_{max}$, and $G_{sum}$, which take the element-wise mean, max, and sum, respectively, over each embedding dimension in the target, and $G_{first}$, which only uses the first target token as the aggregate. The performance of these aggregation functions can be found in Figure 4.

## 5 Argument Identification

Argument identification, also referred to as frame element identification, is defined as the task of finding the frame elements in a sentence related to a particular frame. Given a frame $f$ and a sequence of tokens $x = (x_1, ..., x_n)$, a model detects the frame element $e \in E_f \cup \{none\}$ for each token $x_i \in x$, where $E_f$ is the set of frame elements defined in frame $f$. Since not every token is necessarily a frame element, it is important that *none* is a possible classification.

### 5.1 Model

Our argument identification model, Hydra-A, consists of a separate classification head that classifies the frame elements in each given frame $f$.

Given an input sentence $s$, the tokenized sequence $x$ is produced using BERT's word-piece tokenizer. This tokenizer frequently breaks a single word into multiple tokens, so to prevent tokens belonging to a single word from being classified into different frame elements, our model generates partitions $P = (p_1, ..., p_n)$, where $p_i$ corresponds to the partition which token $x_i$ belongs to (Figure 5).

Our model computes $H$-dimensional[3] contextu-

[3] We use BERT$_{\text{base}}$ in our model which has $H = 768$.

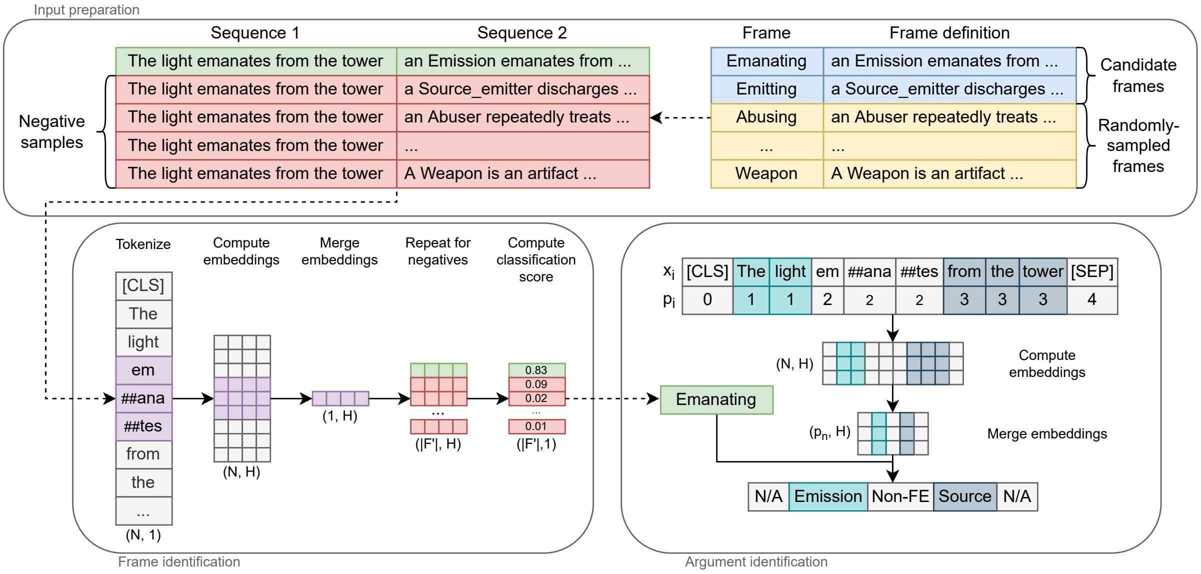

Figure 5: Overview of the entire frame-semantic parsing model on the sentence “The light emanates from the tower". In the top right, the model selects the definitions from candidate frames (blue) and randomly-sampled negative frames (yellow) as detailed in Section 4.3. The model then creates a multiple-choice task using the true sentence-frame definition pair (green) and the negative pairs (red) and passes them along with the target (purple) through the frame identification model. The model computes the merged embedding of the target and computes the classification score for each pair, selecting the highest scoring pair as the predicted frame. This predicted frame is then used by the argument identification model to predict the frame elements in the sentence as detailed in Section 5.

alized BERT embeddings $x^{emb} = (x_1^{emb}, ..., x_n^{emb})$ for each token $x_i \in X$, as done in previous sections. Then token embeddings $x_i^{emb}$ for the same partition (i.e., $p_i = j$ for $j = 0 \rightarrow p_n$) are aggregated using the $G_{mean}$ method discussed in Section 4.3, resulting in a matrix $X^{part}$ of size $(p_n, H)$. This can be seen more clearly in Figure 5.

Our model defines a separate classification head with weights $W_f$ of size $(H, |E_f|)$ for each frame $f$. Each head computes a score $Z^f$ for each partition in the sentence, and only the heads for the frames evoked in the sentence are used.

$$Z^f = X^{part} \times W_f \tag{4}$$

The predicted class for partition $i$ is the one with the highest classification score, i.e., $arg\,max(Z_i^f)$. The weights $W_f$ are learned by minimizing the cross-entropy loss:

$$L = -\sum_{j=1}^{|E_f|} \log \frac{\exp(Z_{i,j}^f)}{\sum_{k=1}^{|E_f|} \exp(Z_{i,k}^f)} \tag{5}$$

Rather than learning a representation of each frame, as is done by Open-SESAME, we approach this problem by defining a separate classification head for each frame $f$ which computes a score for each frame element $e \in E_f$. We make the choice to use many different classification heads so that the weights for each frame's argument identifier are separated from one another, allowing the model to retain its previously trained weights when new frames are introduced, e.g., the fact-checking-related frames introduced by Arslan et al., 2020.

# 6 Experiment Setup

## 6.1 FrameNet 1.7 Dataset

We maintain consistency with previous works in partitioning our training and test sets according to the split used by Open-SESAME and SEMAFOR. In the training split of the dataset, we observe a total of 19,473 sentence-frame pairs belonging to 755 distinct frames. Of these 19,473 pairs, only 18,958 have frame element annotations. Additionally, 1,115 (5.73%) of the 19,473 pairs, belonging to 233 distinct frames, do not have their frames in the test set. In the test set, we observe 6,462 sentence-frame pairs belonging to 563 distinct frames, 6,238 of which have frame element annotations. In the test set, 74 (1.15%) sentences belonging to 41 frames that do not appear in the training set and are thus not learned by the model.

| Model | $P_M$ | $R_M$ | $F_M$ | $P_w$ | $R_w$ | $F_w$ | Acc | Amb |
|---|---|---|---|---|---|---|---|---|
| Open-SESAME | - | - | - | - | - | - | 0.884 | - |
| Peng et al. | - | - | - | - | - | - | 0.900 | - |
| KGFI | - | - | - | - | - | - | **0.924** | **0.844** |
| FIDO (Full model) | **0.824** | **0.805** | **0.801** | 0.931 | **0.922** | **0.920** | 0.922 | 0.838 |
| FIDO (LU-only) | 0.804 | 0.796 | 0.788 | 0.924 | 0.909 | 0.910 | 0.909 | - |
| Hydra-F | 0.817 | 0.804 | 0.800 | **0.932** | 0.921 | **0.920** | 0.923 | 0.841 |

Table 1: Macro and weighted precision, recall, and F1 score for frame identification models. Micro-averaged precision, recall, and F1 score have not been reported as they are equivalent to the accuracy on all targets (Acc). Metrics which are not reproducible due to the inaccessibility of the model have been left blank.

| Model | P | R | F |
|---|---|---|---|
| SEMAFOR | 0.900 | 0.708 | 0.792 |
| Hydra-T | **0.956** | **0.950** | **0.953** |

Table 2: Performance of our target identification model compared against SEMAFOR.

| N | Acc | $F_M$ |
|---|---|---|
| 6 | $0.921 \pm .002$ | $0.800 \pm 0.011$ |
| 8 | $0.918 \pm .001$ | $0.800 \pm 0.005$ |
| 10 | $0.919 \pm .002$ | $0.802 \pm 0.007$ |

Table 3: Accuracy and macro F1 score of our frame identification model using different values for the number of negative samples $N$. Confidence intervals are defined at 95% confidence.

### 6.2 Metrics

One significant flaw in most of the previous works related to frame-semantic parsing is the overuse of accuracy and other micro-averaged metrics which do not account for data imbalance when reporting performance. FrameNet has a very skewed distribution of frames due to the annotations coming from few, similar corpora. As shown in Figure 3, the number of frames that are evoked in less than 5 sentences is roughly 750, or 61% of all of the frames. Because of this heavy imbalance, metrics such as accuracy are misleading for evaluating the performance of models on FrameNet. We believe that macro-averaged performance metrics are the most important as it shows the performance of the model across all frames equally as opposed to being biased towards common frames in FrameNet.

**Target identification** Since the target identification dataset is more balanced and target identification is a binary classification task, macro-averaged metrics are not as necessary as other tasks. To compare our model with SEMAFOR, we use micro-averaged precision, recall, and F1 score. This comparison can be found in Table 2.

**Frame identification** For frame identification, we compute all standard metrics as shown in Table 1, but, as mentioned above, we emphasize the macro-averaged performance of the models. The number of negative samples, $N$, was empirically chosen as 6 based on average model performance at several different $N$ values, as shown in Table 3. More specifically, for each $N$ value, models are trained 5 times and their performance was averaged. We also report our model's accuracy on ambiguous targets (column "Amb" in Table 1) to provide a fair comparison with KGFI and FIDO. Ambiguous targets are targets that can evoke more than one frame.

**Argument identification** For argument identification, we compute performance metrics averaged across all frames using the following equation:

$$\overline{M} = \frac{1}{|F|} \sum_{f \in F} M(X_f, Y_f) \qquad (6)$$

where $M$ is the performance metric (micro, macro, and weighted precision, recall, or F1 score), $X_f$ is the predicted frame elements of each sentence which evokes frame $f$, and $Y_f$ is the corresponding ground-truth. $M$ is averaged over all annotated sentences belonging to each frame.

### 6.3 Baselines

**Target identification** The state-of-the-art target identification model, SEMAFOR, reports results on the SemEval'07 dataset, a dataset which is no longer available, making a direct performance comparison difficult. The SemEval'07 dataset is a derivative of a previous version of FrameNet, so the results should be at least loosely comparable. These results can be found in Table 2.

**Frame identification** We compare our model with several models trained and evaluated on the same partitions of the FrameNet 1.7 dataset. In particular, we compare our model with KGFI, open-SESAME, FIDO, and the work done by Peng et al., 2018 as these are the most recent or most prominent models for frame identification.

**Argument identification** For argument iden-

| Model | $\overline{P_M}$ | $\overline{R_M}$ | $\overline{F_M}$ | $\overline{P_m}$ | $\overline{R_m}$ | $\overline{F_m}$ | $\overline{P_w}$ | $\overline{R_w}$ | $\overline{F_w}$ |
|---|---|---|---|---|---|---|---|---|---|
| Open-SESAME | 0.567 | 0.564 | 0.541 | 0.860 | **0.834** | **0.845** | **0.840** | **0.834** | **0.823** |
| Hydra-A, no examples | 0.654 | 0.677 | 0.669 | 0.801 | 0.776 | 0.848 | 0.778 | 0.776 | 0.811 |
| Hydra-A, no partitioning | 0.651 | 0.659 | 0.643 | 0.843 | 0.807 | 0.821 | 0.789 | 0.807 | 0.789 |
| Hydra-A | **0.700** | **0.715** | **0.693** | **0.866** | 0.828 | 0.843 | 0.819 | 0.828 | 0.814 |

Table 4: Average (over all frames) macro, micro, and weighted precision, recall, and F1 score for argument identification models. Hydra-A outperforms Open-SESAME on macro-averaged metrics, which are more important since they are better suited for imbalanced datasets like FrameNet, while performing on par in other metrics.

tification, we compare our model with Open-SESAME, the current state-of-the-art, on ground truth frames and report the results in Table 4. We also exmaine the performance gained by utilizing exemplar sentences defined in FrameNet to improve the performance on frames with few frame element annotations.

## 7 Results

**Aggregation method test** We perform a test to determine which aggregation method (as explained in Section 4.3) should be used in the model (Figure 4). We observe $G_{mean}$ and $G_{sum}$ perform noticeably better than the $G_{max}$ and $G_{first}$. Summing embeddings can lead to larger values which are notoriously unstable in neural networks. Additionally, by taking the mean of the embeddings, our aggregate embeddings will be less sensitive to outliers and overfitting. Hence, we chose to use $G_{mean}$ in our final model.

**Target identification** Similar to (Das et al., 2014), we find that directly using a statistical model to predict targets results in poor performance. Our target identification model, Hydra-T, correctly identifies targets in the fulltext annotations with a micro-averaged F1 score of 0.953, an improvement of +0.161 over SEMAFOR (Table 2).

**Lexicon filtering** We find that about 81% of the lexical units defined in FrameNet 1.7 evoke only one frame. As a result of this, 54% of the annotated sentence-target pairs have lexical units which can only evoke a single frame. These lexical units trivialize frame identification. Therefore, we follow the practices in KGFI, FIDO, and SEMAFOR by also reporting the performance of our model on the ambiguous targets (targets which have lexical units that evoke 2 or more frames). As seen in Table 1 (column "Amb"), ambiguous targets are harder to classify, resulting in a reduction in performance compared to the full test set.

**Frame identification** As shown in Table 1, Hydra-F provides significant improvements over FIDO's LU-only model, which we use to show the improvement our training method provides over FIDO's. Additionally, Hydra-F has a classification accuracy comparable to KGFI and FIDO's full model, without the frame definitions used by FIDO or the frame-semantic structure used by KGFI. This implies that our training method allows our model to learn to classify the frames evoked by targets with significantly less resources.

**Argument identification** We use Equation 6 to compute the performance measures reported in Table 4. To ensure consistency and robustness in future comparisons, we recommend that future works also produce these results. We observe a significant improvement with Hydra over Open-SESAME on all macro-averaged metrics while being on par on other metrics. Based on this observation, our model is able to learn to predict less-common frames better than Open-SESAME.

To show the importance of effective sentence partitioning, we also compare our model using partitions based on the ground-truth frame elements with the same model without partitioning (i.e., frame elements are predicted for each token in the sentence). As shown in Table 4, there is a significant increase in performance when using effective partitioning, thus showing the importance of a high-quality partitioning algorithm.

## 8 Conclusion

This paper presented a new frame-semantic parsing system. We showed how using a generation-then-filtering method with a transformer-based model can reach state-of-the-art performance on target identification. The system's frame identification model attains classification performance on par with state-of-the-art solutions with significantly less resources. We show that using separate classification heads for each frame allows our argument identification model to reach state-of-the-art performance. We also argue for the use of macro-averaged performance metrics due to the heavy class imbalance in FrameNet.

## Limitations

Our argument identification model has two main limitations. The first is our use of ground-truth frame element partitions. It is vital to the performance of our argument identification model to be able to partition words which belong to the same frame element together. Previous works, such as Open-SESAME and SEMAFOR use methods such as dependency parsers and constituency parsers to partition the texts; however, these methods are not fully robust and only cover about 80% of the frame elements in the fulltext annotations.

The second limitation is that the model is not able to transfer learned embeddings across frames. For instance, if two frames each have a “Time" frame element, the weights corresponding to each frame will have different representations of the “Time" frame element since the losses are only computed for the particular frame evoked. This is not ideal since most frames have very little training data for each frame element.

Given more resources, we would like to have experimented with more values for the number of negative samples $N$ and different sampling methods.

Our work was done based on the original FrameNet and under the assumption that the inputs are in English. We did not consider other versions of FrameNet, such as the Chinese FrameNet, French FrameNet, or Japanese FrameNet while developing our model and cannot ensure the linguistic tools we used will be applicable to other languages.

## Ethics Statement

We have not taken the opportunity to assess the existence of biases in the FrameNet dataset which could inadvertently bias our system. Applications of our system could lead to potential biases for or against certain genders, races, or ethnicities due to underlying biases in the training data. In particular, it is known that FrameNet uses annotations from materials which mention chemical and nuclear weapons and regions such as North Korea, Iran, Syria, and Taiwan.

We believe further improvement on frame-semantics can produce more explainable results for down-stream applications and assist in projects which have positive societal impacts.

## References

Fatma Arslan, Josue Caraballo, Damian Jimenez, and Chengkai Li. 2020. Modeling factual claims with semantic frames. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2511–2520, Marseille, France. European Language Resources Association.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. 2014. Frame-semantic parsing. *Computational linguistics*, 40(1):9–56.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. 2010. Probabilistic frame-semantic parsing. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 948–956.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Charles J Fillmore et al. 2006. Frame semantics. *Cognitive linguistics: Basic readings*, 34:373–400.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Tianyu Jiang and Ellen Riloff. 2021. Exploiting definitions for frame identification. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2429–2434, Online. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018. Learning joint semantic parsers from disjoint data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1492–1502, New Orleans, Louisiana. Association for Computational Linguistics.

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

Anders Søgaard, Barbara Plank, and Héctor Martínez Alonso. 2015. Using frame semantics for knowledge extraction from twitter. In *AAAI Conference on Artificial Intelligence*.

Evangelia Spiliopoulou, Eduard Hovy, and Teruko Mitamura. 2017. Event detection using frame-semantic parser. In *Proceedings of the Events and Stories in the News Workshop*, pages 15–20, Vancouver, Canada. Association for Computational Linguistics.

Xuefeng Su, Ru Li, Xiaoli Li, Jeff Z. Pan, Hu Zhang, Qinghua Chai, and Xiaoqi Han. 2021. A knowledge-guided framework for frame identification. In *ACL*.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *ArXiv*, abs/1706.09528.

# A Reproducibility

## A.1 Hyperparameters

Our frame-semantic parser is built using PyTorch with the Hugging Face library[4]. We use Hugging Face's implementation of the pretrained BERT-Base model.

### A.1.1 Target Identification Model

We optimize the weights of our target identification model for 20 epochs using AdamW (Loshchilov and Hutter, 2017) with a learning rate of $1e-5$ and a batch size of 1. Each batch consists of a single sentence and all of its candidate targets, and the gradients are accumulated over 4 batches, resulting in an effective batch size of 4.

### A.1.2 Frame Identification Model

We optimize the weights of our frame identification model for 5 epochs using AdamW with a learning rate of $2e-5$ and a batch size of 16. To ensure we have sufficient space for the input sequence, frame definition, and lexical unit definition, we use a maximum sequence length of 300 tokens.

### A.1.3 Argument Identification Model

We optimize the weights of the pretrained BERT layers using AdamW with a learning rate of $1e-5$, and optimize the weights of each classification head using AdamW with a learning rate of $1e-3$. We choose these values intuitively as we do not want to make very large changes to the BERT layers, instead opting for larger changes to the weights of each classification head. All other parameters are consistent between the BERT and classification layers, i.e., $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$, and a weight decay coefficient of $1e-2$. Additionally, our argument identification model was trained for 20 epochs with a batch size of $8$, keeping the model checkpoint with the highest macro F1 score on the test set. For the results reported in this paper, we used the checkpoint at 4 epochs.

[4] The Hugging Face library can be found at https://huggingface.co/