Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 1671–1679
Marseille, 20-25 June 2022

Combining ELECTRA and Adaptive Graph Encoding for Frame Identification

Fabio Tamburini
FICLIT-University of Bologna, Italy
fabio.tamburini@unibo.it

Abstract

This paper presents contributions in two directions: first we propose a new system for Frame Identification (FI), based on pre-trained text encoders trained discriminatively and graphs embedding, producing state of the art performance and, second, we take in consideration all the extremely different procedures used to evaluate systems for this task performing a complete evaluation over two benchmarks and all possible splits and cleaning procedures used in the FI literature.

Keywords: frame semantics, frame identification, evaluation

1. Introduction

Frame Semantics (Fillmore, 1982; Fillmore and Baker, 2010) is one of the more successful semantic theory and FrameNet[1] (Fillmore et al., 2003) is an invaluable resource both for linguistic and computational analyses. A semantic frame represents an event or scenario, and possesses frame elements that participate in the event/scenario with different roles.
On the computational side, considering the Frame Semantic Parsing task, studies in literature concentrated on three major subtasks (Das et al., 2014):

- *target identification*, involving the selection of a word token, or word token sequences (the target predicate), evoking some frame in a given sentence. In frame semantics, verbs, nouns, adjectives, and even prepositions can evoke frames under certain conditions;
- *frame identification* aims at finding the exact frame evoked by a target predicate in a given sentence, a sort of word-sense disambiguation task using frames as senses;
- *argument identification* (a.k.a. semantic role labeling) is the task of identifying words and phrases as the appropriate arguments of the target predicate, the so called Frame Elements, and their specific roles.

See Figure 1 for some examples and a more detailed explanation of the three tasks and a brief description of FrameNet frame-to-frame relation structure.

In this paper, we focus our attention on the Frame Identification (FI) task. There is a long series of studies in literature for solving the FI task (Das and Smith, 2011; Das et al., 2014; Hermann et al., 2014; Swayamdipta et al., 2017; Hartmann et al., 2017; Yang and Mitchell, 2017; Botschen et al., 2018; Peng et al., 2018; Sikos and Padó, 2019; Popov and Sikos, 2019; Tan and Na, 2019; Chen et al., 2021; Jiang and Riloff, 2021; Su et al., 2021), both using traditional techniques and various neural network approaches.

Examining in detail the most recent literature presenting state of the art systems based on Deep Neural Networks, we found four main studies.

Tan and Na (2019) proposed a very simple method based on transformers (Vaswani et al., 2017; Devlin et al., 2019) and positional attention obtaining good performance results.

The solution by Chen et al. (2021) consists of four modules, a Bidirectional-LSTM encoder module for encoding the input sentence and three decoder modules, for solving respectively frame identification, argument identification and role classification, jointly optimized for solving all these subtasks of frame semantic parsing together.

Jiang and Riloff (2021) exploited lexical unit and frame definitions concatenated with target sentences for getting BERT embeddings (Devlin et al., 2019) and estimate the probability of a given frame to be the correct frame evoked by the input target.

The work from Su et al. (2021) is very recent, it has been developed independently at the same time of this work and share some similar ideas employing graph embeddings and pre-trained language models. Our work, however, is based on a more reliable graph embedding technique (see Sect. 2.1) and it has been extensively evaluated over the whole set of benchmark configurations (see Sect. 3).

The contribution of our paper is two-fold: first, we present a new system for FI producing state of the art performance (Sect. 2) and, second, we take in consideration all the extremely different procedures used to evaluate systems for this task performing a complete evaluation over the two benchmark datasets and all possible splits and cleaning procedures used in the FI literature (Sect. 3 and 4). Sect. 5 and 6 analyse in more detail some aspects of the proposed system and the obtained results and Sect. 7 draws some conclusions.

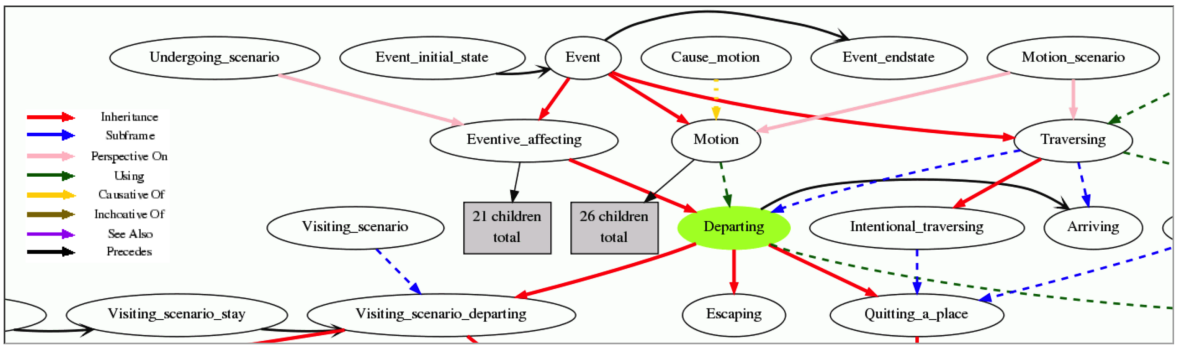
[1] https://framenet.icsi.berkeley.edu/fndrupal/

We hurried down the village street and found, as we had expected, that [the inspector]$_{Theme}$ was [just]$_{Time}$ **LEAVING**$_{[\text{DEPARTING}]}$ [his lodgings]$_{Source}$.
Mysteriously, the Anasazi vanished from the valley around a.d. 1150, **LEAVING**$_{[\text{CAUSATION}]}$ [it]$_{Affected}$ [to be repopulated by the Southern Paiutes, another hunter-gatherer tribe]$_{Effect}$.
True to wild-West stereotypes, Stewart was slain by a neighboring farmer, **LEAVING**$_{[\text{ABANDONMENT}]}$ [his strong-willed wife, Helen]$_{Theme}$, [to assume the duties of the ranch]$_{Explanation}$.

(a)

[We]$_{Self\_mover}$ **HURRIED**$_{[\text{SELF\_MOTION}]}$ [down the village street]$_{Path}$ and found, as we had expected, that the inspector was just leaving his lodgings.
We hurried down the [village]$_{Relative\_location}$ [**STREET**]$_{Roadway[\text{ROADWAYS}]}$ and found, as we had expected, that the inspector was just leaving his lodgings.
[We]$_{Cognizer}$ hurried down the village street and **FOUND**$_{[\text{BECOMING\_AWARE}]}$, as we had expected, [that the inspector was just leaving his lodgings]$_{Phenomenon}$.
We hurried down the village street and found , as [we]$_{Cognizer}$ had **EXPECTED**$_{[\text{EXPECTATION}]}$, [that the inspector was just leaving his lodgings]$_{Phenomenon}$.
We hurried down the village street and found, as we had expected, that [the inspector]$_{Theme}$ was [just]$_{Time}$ **LEAVING**$_{[\text{DEPARTING}]}$ [his lodgings]$_{Source}$.

(b)



(c)

Figure 1: (a) Three examples of the target lemma *leave.v* from FrameNet 1.7 each evoking a different frame. '**LEAVING**' represents the target word evoking the various frames, namely DEPARTING, CAUSATION and ABANDONMENT, and the text fragments in square brackets represent the Frame Elements with their respective role. (b) In FrameNet full-text annotations we could have more than one target word evoking different frames in the same sentence, each of which is a FI problem to be solved. In the example, five target words are present in the same sentence, evoking different frames and involving various FEs. (c) A fragment of the FN 1.7 graph connecting frames with various frame-to-frame relations centered on the frame DEPARTING as showed by the Frame Grapher from the original FN Web site.

2. System Architecture

Formally, the FI task can be described in this way: let $\mathcal{S} = w_1, w_2, ...w_n$ represent the actual sentence with a marked predicate t, the target, that evokes a member of the set of all possible frames $\mathcal{F} = \{f_1, ..., f_k\}$, built by extracting data from FrameNet (FN) for a given FN version. FI is generally seen in literature as a classification task. For complete information about this long-standing task please refer also to (Das et al., 2014; Swayamdipta et al., 2017).

The proposed architecture for solving FI is depicted in Figure 2; it is composed of various modules described in detail in the following subsections.

2.1. Attributed Graph Embeddings

We relied on the AGE proposal from Cui et al. (2020) for embedding part of the FrameNet graph and, in particular, for deriving highly informative frame embeddings containing both frame features and FN graph information.

As discussed in (Yang et al., 2021), most of existing network embedding methods rely on Graph Convolutional Networks (GCN) (Kipf and Welling, 2017) and are based on graph autoencoder (GAE) and variational graph autoencoder (VGAE) (Kipf and Welling, 2016). These GCN-based methods have some major drawbacks described in detail in (Yang et al., 2021): (a) the entanglement of the filters and weight matrices composing GCNs provides no performance gain for semi-supervised graph representation learning, and even harms training efficiency since it deepens the paths of back-propagation; (b) considering the graph convolutional filters, that act as Laplacian smoothing filters applied on the feature matrix for low-pass de-

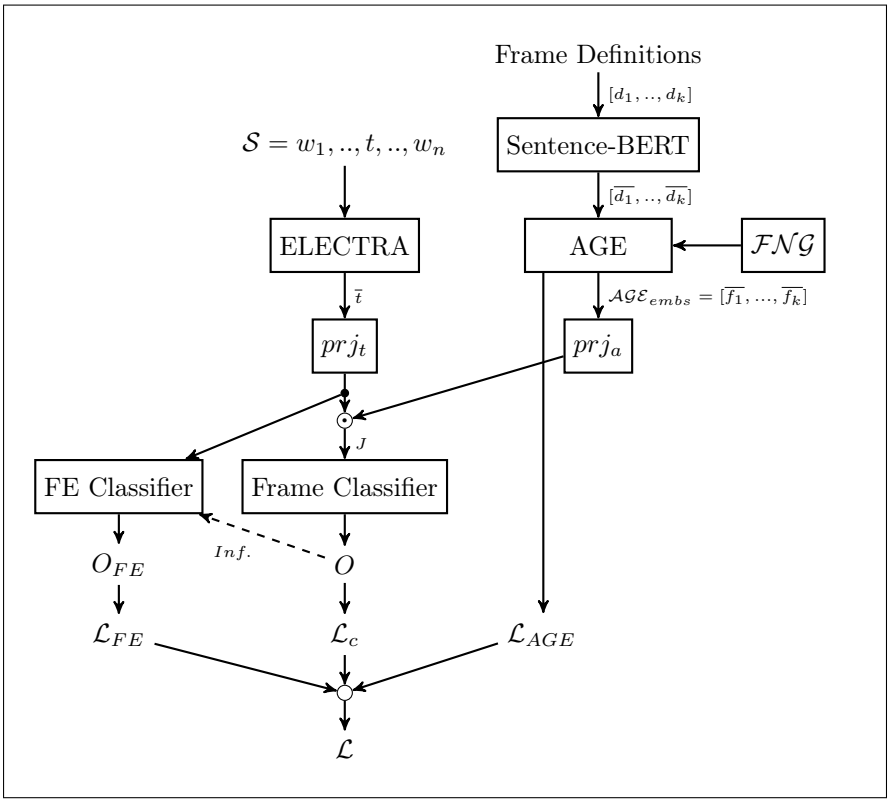


Figure 2: The various modules composing the full 'Electra-AGE_FE' system for Frame Identification. The detailed description of all components can be found in Section 2.

noising, it can be shown that they are not optimal since they cannot filter out noises in some high-frequency intervals not reaching the best smoothing effect; (c) training objectives of these algorithms, either reconstructing the adjacency matrix or feature matrix, present some drawbacks: reconstructing the adjacency matrix sets it as the ground truth pairwise similarity, while it is not appropriate for the lack of feature information. Recovering the feature matrix will force the model to remember high-frequency noises in features, and thus is inappropriate as well (Yang et al., 2021).

For all these reasons, Cui et al. (2020) proposed Adaptive Graph Encoder (AGE), a framework for attributed graph embedding composed of a nonparametric Laplacian filter to perform low-pass filtering in order to get smoothed node features and an adaptive encoder to learn better node embeddings. The reconstruction training objectives has been replaced with adaptive learning which selects training samples from the pairwise similarity matrix and fine tunes the embeddings iteratively. AGE is able to embed a graph structure with feature-rich nodes into an embedding space in which node embeddings contain both original node features and graph-structure information.

Considering the discussion above and contrary to the choices made by Su et al. (2021) that applied CGN-based methods, we preferred to adopt the AGE proposal from Cui et al. (2020) for deriving structurally-informed frame embeddings. We first extract frame embeddings $\mathcal{D}_{embs} = [\overline{d_1}, ..., \overline{d_k}] \in \mathbb{R}^{k \times d}$ from FrameNet frame definitions $\mathcal{D} = \{d_1, ..., d_k\}$, with d_i the definition of f_i, by using the Sentence-BERT model (Reimers and Gurevych, 2019), for all the k frames in a given FN version and use them as starting node features for the AGE model. d represents the dimension of Sentence-BERT embeddings. Then, we extract the FrameNet subgraph $\mathcal{FNG}$ projected by the *Inheritance*, *Subframe* and *Using* frame relations from FN, as in Popov and Sikos (2019), and use them as graph-structure input in the AGE model for obtaining the structurally-informed frame embeddings $\mathcal{AGE}_{embs} = [\overline{f_1}, ..., \overline{f_k}] \in \mathbb{R}^{k \times y}$, where y is the AGE output embedding dimension.

2.2. Classifiers

The input sentence $\mathcal{S}$ is processed by one contextual-embedding technique based on transformers (Vaswani et al., 2017) for obtaining all the contextualised word-embeddings for each word in $\mathcal{S}$ and, as a consequence, also for the target t, namely $\bar{t}$ (for multi-word expression targets we averaged the corresponding embeddings). In this study we relied on ELECTRA, recently proposed by Clark et al. (2020), a text encoder pre-trained with a more sample-efficient task called 'replaced token detection' as a discriminator rather than a generator in a set-up resembling a generative adversarial network.

The AGE and target embeddings were combined by

first projecting them into a common joint target-label space of dimension h by two linear layers with bias, namely prj_t and prj_a, and then, by using the component-wise multiplication '$\odot$', into

$$J = [prj_a(\overline{f_1}) \odot prj_t(\overline{t}), ..., prj_a(\overline{f_k}) \odot prj_t(\overline{t})] \in \mathbb{R}^{k \times h}$$

to measure the compatibility between the target and each possible frame as done in Pappas and Henderson (2019). In order to get the final classification probabilities over all possible frames we computed $O_i = J_i \cdot W_i + B_i, i = 1, ..., k$, where $W \in \mathbb{R}^{k \times h}$, $B \in \mathbb{R}^k$ and '$\cdot$' is the dot product, performing the equivalent of k linear $h \times 1$ layers with bias applied row-wise to J, and then applied the softmax function.

In order to enrich the information injected into the model, we trained a further multi-label classifier, in the spirit of multi-task learning, for predicting the Frame Elements (FE) involved in a specific frame invocation; from the projected target embedding, through a further linear layer with bias P, we obtain the output $\overline{O_{FE}} = sigmoid(P(prj_t(\overline{t}))) \in \mathcal{R}^f$, with f the total number of FE for a given FN version.

The whole system does not require a very big model consisting in only 1.6 million parameters.

2.3. Optimisation, Losses and Inference

We optimise our model by using the AdamW algorithm (Loshchilov and Hutter, 2019) with respect to three losses: (a) the standard loss for link prediction proposed by (Cui et al., 2020), $\mathcal{L}_{AGE}$, for the AGE model; (b) the cross-entropy loss, $\mathcal{L}_c$, for the classifier output $\overline{O}$ and (c) the binary cross-entropy loss, $\mathcal{L}_{FE}$, for the multi-label FE classifier output $\overline{O_{FE}}$. We combine these losses and optimise the model by using the joint loss

$$\mathcal{L} = \gamma_2 \left[\gamma_1 \mathcal{L}_c + (1 - \gamma_1) \mathcal{L}_{AGE}\right] + (1 - \gamma_2) \mathcal{L}_{FE}$$

with $\gamma_1, \gamma_2 \in [0, 1]$.

With regard to inference on output frames, we restrict the set of possible frames to all those potentially evoked by the target t. For FE classification during inference, the final decision about the FEs involved in a specific problem instance are restricted to the set of all possible FEs being part of every frame potentially evoked by t, thresholding the logits on 0.0 for taking the decision.

3. Experimental setting

As noted by Kabbach et al. (2018), evaluating FI systems is a very complex task: despite the availability of two standard benchmarks adopted in current literature, namely FN 1.5 and FN 1.7, examining the large bundle of works devoted to FI we found a lot of different procedures based on only one of the two FN versions and on different splits and cleaning criteria.

The older works tested the proposed systems on the full-text annotations of FN 1.5, considering the split first introduced by Das and Smith (2011), containing 23 documents for the test set with 4,458 predicates. From the remaining 55 documents, 16 documents were chosen as validation set following Hermann et al. (2014) (we call this split 'Val16'). Swayamdipta et al. (2017), accepting the same split for the test set, proposed instead a different split in which only 8 documents were included into the validation set (we call it 'Val8'). Kabbach et al. (2018), adopting the same split as Swayamdipta et al. (2017), noted that data were not clean, some sentences or annotations were inserted twice or more times across splits and developed the `pyfn` package[2] to clean them and to guarantee no duplicates and no overlaps within and across splits and, thus, to perform more correct evaluations (we call this split 'Val8C').

With regard to FN 1.7, used by more recent studies, problems are exactly the same: some works used the Das and Smith (2011) split, others the Swayamdipta et al. (2017) split and, again, data has to be cleaned in order to have a more correct and reliable benchmark (Kabbach et al., 2018). We use the same split names introduced before for FN1.5.

Moreover, some studies used as further training data also the FN exemplar sentences annotated partially for showing the use of the various FEs, some claiming that they improve the results (Yang and Mitchell, 2017; Chen et al., 2021) while others stating that they, being only partially annotated, could confuse the systems classification process (Das et al., 2014). Actually, there is also the problem that some exemplar sentences have been included into validation and test sets, especially for FN 1.7, and thus cannot definitely be used for training without applying some cleaning procedure. It is worth noting that Tan and Na (2019) only partially clean data, removing duplicates only in the exemplars added to the training set, but, as they do not provide any procedure or instance count for their splits, we cannot reliably reproduce their evaluation and thus we will not consider their work for comparisons.

Given this very problematic panorama, comparing our results with the literature and the state of the art is very complex and, choosing a specific benchmark, split and cleaning option, only a subset of the interesting studies can be reliably considered. In our opinion this is not acceptable, thus we decided to test our system in any configuration and compare the obtained results with the appropriate studies, giving a complete and systematic picture of the FI results in literature for any considered configuration and reliable conclusions about the effectiveness of the proposed solution.

We relied on the package `pyfn` cited before to extract data both in uncleaned and cleaned form from the original annotations in FN releases, with or without the additional exemplar sentences and annotations. Table 1 shows the number of FI instances in each split

[2] `https://github.com/akb89/pyfn`

and configuration used by some work in literature and then tested here as well as the number of ambiguous instances in the various test sets.

Benchmark Configuration	# FI Instances		
	Train	Valid.	Test (Ambig.)
FN1.5_Val16	15044	4436	4458 (2025)
FN1.5_Val8	17148	2332	4458 (2025)
FN1.5_Val8C	16706	2319	4148 (1850)
FN1.5_Val8C†	170889	2319	4148 (1850)
FN1.7_Val8	19903	2309	6728 (3293)
FN1.7_Val8C	19550	2309	6446 (3114)
FN1.7_Val8C†	192554	2309	6446 (3114)

Table 1: All combinations of splits, cleaning (C) and use of exemplars in the training set (†) from FN versions considered in literature and tested here.

In Table 2 we listed the pre-trained models and hyperparameters used for testing our system without having optimised them in any way.

Pre-trained model	Dim.
SBERT: stsb-roberta-base	768
ELECTRA: electra-base-discriminator	768

Hyperpar.	Value	Hyperpar.	Value
d, y	768	γ_1	0.5
h	256	γ_2	0.1

Table 2: Pre-trained models and hyperparameters used in the proposed system.

The influential paper from Reimers and Gurevych (2017) makes clear to the community that reporting a single score for each DNN training/evaluation session could be heavily affected by the system random initialisation and we should instead report the mean and standard deviation of various runs, with the same setting, in order to get a more accurate picture of the real systems performance and make more reliable comparisons between them. Despite this, most of the cited literature for FI still present the results as a single score without explicitly state if it is the average, the maximum value or any other combination of various experiments. A notable exception in recent literature regards the work from Jiang and Riloff (2021) that presents the average of three runs.

For these reasons, any new result proposed in this paper is presented as the mean and standard deviation of FI accuracy (the standard metric for this task) over 10 runs with different random initialisations. In this way, we should give a real picture of our system performances.

4. Results

Table 3 shows the performance of the proposed system, Electra-AGE_FE, compared to any work found in literature w.r.t. the different benchmark configurations in Table 1. Our system consistently outperform any other proposal in a highly significant way except when

Model	FN1.5_Val16		
	All	Amb.	Max
(Das et al., 2014)	83.60	69.19	-
(Popov and Sikos, 2019)	87.03	72.48	-
(Hartmann et al., 2017)	87.63	73.80	-
(Yang and Mitchell, 2017)†	88.20	75.70	-
(Hermann et al., 2014)	88.41	73.10	-
(Botschen et al., 2018)	88.82	75.28	-
(Sikos and Padó, 2019)	91.26	80.77	-
(Jiang and Riloff, 2021)	91.30	81.00	-
Electra-AGE_FE	**91.71**$^{\bullet}$ ±0.17	**82.01**$^{\bullet}$ ±0.37	**91.97/ 82.56**

Model	FN1.5_Val8		
	All	Amb.	Max
(Swayamdipta et al., 2017)	87.51	-	-
(Peng et al., 2018)†	90.00	78.00	-
(Chen et al., 2021)†	90.50	79.10	-
(Su et al., 2021)	92.13	82.34	-
Electra-AGE_FE	**92.21**$^{-}$ ±0.12	**83.13**$^{\bullet}$ ±0.23	**92.42/ 83.56**

Model	FN1.5_Val8C		
	All	Amb.	Max
(Kabbach et al., 2018)	83.20	73.60	-
(Kabbach et al., 2018)†	84.60	69.30	-
Electra-AGE_FE	**92.57**$^{\bullet}$ ±0.12	**83.58**$^{\bullet}$ ±0.27	**92.79/ 84.05**
Electra-AGE_FE†	92.56$^{\bullet}$ ±0.09	83.54$^{\bullet}$ ±0.20	92.67/ 83.79

Model	FN1.7_Val8		
	All	Amb.	Max
(Peng et al., 2018)†	89.10	77.50	-
(Jiang and Riloff, 2021)	92.10	83.80	-
(Su et al., 2021)	**92.40**	**84.41**	-
Electra-AGE_FE	92.24° ±0.14	84.26^{-} ±0.29	**92.46/ 84.73**

Model	FN1.7_Val8C		
	All	Amb.	Max
(Kabbach et al., 2018)	82.30	70.00	-
(Kabbach et al., 2018)†	83.60	66.70	-
Electra-AGE_FE	**92.33**$^{\bullet}$ ±0.13	**84.22**$^{\bullet}$ ±0.27	**92.49/ 84.55**
Electra-AGE_FE†	91.42$^{\bullet}$ ±0.08	82.34$^{\bullet}$ ±0.17	91.62/ 82.76

Table 3: FI accuracy results for 'All' or 'Ambiguous' instances in FN 1.5 and FN 1.7 evaluation benchmarks used in literature and the splits listed in Table 1. 'Max' represents the best performance obtained by our system. Results marked with '†' make use of FN exemplars during training only for Val8C configurations because some exemplars, for the other configurations, have been inserted into the validation and test sets. Some symbols mark the significance using a one-sample t-test when comparing our results with the best found in literature ($^{-}$: p>0.05, *: 0.05>p>0.01, $^{\circ}$: 0.01>p>0.001, $^{\bullet}$: p<0.001).

compared to the work from (Su et al., 2021). When tested on FN1.5_Val8, our system produced a better result but it was not significant (p=0.078) if considering all instances and highly significantly better on ambiguous ones. On FN1.7_Val8, Su et al. (2021) performed significantly better on all instances but not on ambiguous ones. In both cases our system exhibits better performance on ambiguous FI instances. However, it is very relevant to note that the cited work from Su et al. (2021) presented a unique test performance and not the average of multiple runs and it is not clear if this is the best absolute performance or not. Our 'Max' result would be a bit better when compared to their results for FN1.7_Val8 and much better for FN1.5_Val8.

In general, the Val8 split of FN 1.5 produces better results w.r.t. the Val16 split because more data are placed in the training set and even better the results for the Val8C cleaned datasets.
Adding the dataset exemplars to the cleaned splits, the only conditions for which is safe doing it, does not help at all in increasing system performance and this seems to confirm that having only partially annotated sentences, as in the case of FN exemplars, does not give enough information for helping the system in the disambiguation phase. These results seems a bit better for FN 1.5, a version containing less data and more problems in the annotations, but, on the newer FN 1.7 version, being enriched and cleaned a bit more, adding exemplars reduces the system performance by a large margin.

5. Ablation Study

In order to fully understand the contributions of the various system components in Figure 2, we performed an ablation study increasing the system complexity one step at a time. Table 4 shows the results obtained with the various systems when applied to the reference configuration FN1.5_Val8.
The simplest system, 'Electra-T', tries to predict the correct frame using only the ELECTRA embedding for the target word, namely $\bar{t}$: despite the simplicity of this systems its performance is indeed quite good, a bit less that 92%. 'Electra-T_FE' slightly increases the complexity by adding also the FEs prediction component and it improves the performance a bit. A big improvement in system performance was obtained adding the AGE component for including structurally-informed frame embeddings into the game and working in a joint target-label space. The complete system, integrating all modules further improves the performance showing that all the various pieces forming the whole 'Electra-AGE_FE' system really contribute to the final results.

6. FE Classification

Even if this is not the focus of this study, in order to gather more insights about the real effectiveness of the proposed model, we measured also the Frame Element Classification performance.

Model	**FN1.5_Val8**	
	All	Amb.
Electra-T	91.98±0.12	82.62±0.25
Electra-T_FE	92.00±0.11	82.67±0.24
Electra-AGE	92.14±0.11	82.97±0.25
Electra-AGE_FE	**92.21**±0.12	**83.13**±0.23

Table 4: FI accuracy for the various steps considered in the ablation study. The differences between Electra-AGE/Electra-T and Electra-AGE_FE/Electra-T_FE when applying a two-sample t-test are both statistically highly significant (0.01>p>0.001), showing the importance of adding AGE frame embeddings to our model.

A frame typically possesses more than a single FE, thus FEs classification must be configured as a multi-label multi-class classification problem. As described in Sect 2.2 and 2.3, the output of the FE Classifier is the result of a sigmoid function with the decision threshold set to 0.0. This approach is the standard choice for multi-label classifiers.
The Jaccard Score (JS), defined as the size of the intersection divided by the size of the union of two label sets, is used to compare the set of predicted labels for a sample to the corresponding set of true labels and it is thus a good metric for measuring the performance of multi-label classifiers, where JS = 1 means perfect classification and JS = 0 no overlapping between system predictions and true labels.
We took the binary output of the FE Classifier, restricted to the subset of all possible FEs for a given instance (see Sect 6), and compared it with the binary vector of correct labels measuring the degree of overlapping between them by JS. Table 5 shows the means and std.dev. of JS for all the tested split configurations.

Split Configuration	**Electra-AGE_FE Jaccard Score (in %)**
FN1.5_Val16	35.24±0.95 (max 36.20)
FN1.5_Val8	37.30±1.16 (max 39.27)
FN1.5_Val8C	36.64±1.18 (max 37.73)
FN1.5_Val8C†	36.65±1.46 (max 39.38)
FN1.7_Val8	38.39±1.55 (max 39.86)
FN1.7_Val8C	38.34±1.07 (max 39.41)
FN1.7_Val8C†	39.55±1.41 (max 42.77)

Table 5: Electra-AGE_FE Frame Elements Classifier Jaccard Score (in %).

Given that we did not make any effort for tuning the system and increasing FEs classification performances, JSs slightly less than 40% look very promising.

7. Discussion and Conclusions

Most of the studies in literature producing the best results for the various benchmarks, namely Tan and Na (2019) and Su et al. (2021), do not provide the system codes and a precise description of their experimental procedures, thus it is very difficult to reproduce their

results and compare them with ours; when a comparison is possible, our results are in line with their accuracies or better for all benchmarks. Notable exceptions are the works from Kabbach et al. (2018) and Jiang and Riloff (2021) that provide all experimental setting information and allow for a complete and reliable comparison: also in these cases our system obtained the best results.

Kabbach et al. (2018) presents the unique reference on fully cleaned configurations, but it used a very simple and not a really competitive system. We, anyway, preferred to provide a reference also for these split and cleaning configurations because, despite no other recent work is using them, in our opinion they should be adopted as the real references, being the most correct and reliable benchmark configurations among those used in literature.

There are various studies (Litkowski, 2014; O'Hara and Wiebe, 2009; Matsubayashi et al., 2009) proposing to organise the FrameNet FEs into a hierarchy of semantic roles. Adopting such perspective, it could be possible in our future works to approach the FE classification problem applying classification techniques for multi-label hierarchically organised labels (Murty et al., 2018; Wehrmann et al., 2018; Xu and Barbosa, 2018; Mao et al., 2019; Muller and Smith, 2020; Zhou et al., 2020) and improve the classification results. Our future plans regard also the development of a full fledged system embodying also target and argument identification.

All the benchmark configuration datasets and the code for reproducing our results can be downloaded from our GitHub repository[3].

Acknowledgments

We acknowledge the CINECA[4] award no. HP10C7XVUO (project QT4CLML) under the ISCRA initiative, for the availability of HPC resources and support.

8. Bibliographical References

Botschen, T., Gurevych, I., Klie, J.-C., Mousselly-Sergieh, H., and Roth, S. (2018). Multimodal frame identification with multilingual evaluation. 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 1481–1491, New Orleans, Louisiana. Association for Computational Linguistics.

Chen, X., Zheng, C., and Chang, B. (2021). Joint multi-decoder framework with hierarchical pointer network for frame semantic parsing. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2570–2578. Association for Computational Linguistics.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Cui, G., Zhou, J., Yang, C., and Liu, Z. (2020). Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 976–985, New York, NY, USA. Association for Computing Machinery.

Das, D. and Smith, N. A. (2011). Semi-supervised frame-semantic parsing for unknown predicates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1435–1444, Portland, Oregon, USA. Association for Computational Linguistics.

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

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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, June. Association for Computational Linguistics.

Fillmore, C. J. and Baker, C. F. (2010). A frames approach to semantic analysis. In B. Heine et al., editors, *Oxford Handbook of Linguistic Analysis*, page 313–341. OUP.

Fillmore, C. J., Johnson, C. R., and Petruck, M. R. (2003). Background to FrameNet. *International Journal of Lexicography*, 16(3):235–250.

Fillmore, C. J. (1982). Frame semantics. In *Linguistics in the Morning Calm*, pages 111–137. Hanshin Publishing Co., Seoul, South Korea.

Hartmann, S., Kuznetsov, I., Martin, T., and Gurevych, I. (2017). Out-of-domain FrameNet semantic role labeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 471–482, Valencia, Spain. Association for Computational Linguistics.

Hermann, K. M., Das, D., Weston, J., and Ganchev, K. (2014). Semantic frame identification with distributed word representations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1458, Baltimore, Maryland. Association for Computational Linguistics.

Jiang, T. and Riloff, E. (2021). Exploiting definitions for frame identifications. In *Proceedings of EACL 2021*, pages 2429–2434. Association for Computational Linguistics.

[3] https://github.com/ftamburin/Electra-AGE_FE

[4] https://www.cineca.it/en

Kabbach, A., Ribeyre, C., and Herbelot, A. (2018). Butterfly effects in frame semantic parsing: impact of data processing on model ranking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3158–3169, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kipf, T. N. and Welling, M. (2016). Variational graph auto-encoders.

Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations - ICLR '17*.

Litkowski, K. (2014). The framenet frame element taxonomy. Technical report.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Mao, Y., Tian, J., Han, J., and Ren, X. (2019). Hierarchical text classification with reinforced label assignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 445–455, Hong Kong, China, November. Association for Computational Linguistics.

Matsubayashi, Y., Okazaki, N., and Tsujii, J. (2009). A comparative study on generalization of semantic roles in framenet. In *In Proc. ACL*.

Muller, B. R. and Smith, W. A. (2020). A hierarchical loss for semantic segmentation. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020)*, pages 260–267, Valletta, Malta.

Murty, S., Verga, P., Vilnis, L., Radovanovic, I., and McCallum, A. (2018). Hierarchical losses and new resources for fine-grained entity typing and linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 97–109, Melbourne, Australia. Association for Computational Linguistics.

O'Hara, T. and Wiebe, J. (2009). Exploiting semantic role resources for preposition disambiguation. *Computational Linguistics*, 35(2):151–184, June.

Pappas, N. and Henderson, J. (2019). GILE: A generalized input-label embedding for text classification. *Transactions of the Association for Computational Linguistics*, 7:139–155.

Peng, H., Thomson, S., Swayamdipta, S., and Smith, N. A. (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.

Popov, A. and Sikos, J. (2019). Graph embeddings for frame identification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 939–948, Varna, Bulgaria, September. INCOMA Ltd.

Reimers, N. and Gurevych, I. (2017). Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark, September. ACL.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sikos, J. and Padó, S. (2019). Frame identification as categorization: Exemplars vs prototypes in embeddingland. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 295–306, Gothenburg, Sweden, May. Association for Computational Linguistics.

Su, X., Li, R., Li, X., Pan, J. Z., Zhang, H., Chai, Q., and Han, X. (2021). A knowledge-guided framework for frame identification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5230–5240, Online. Association for Computational Linguistics.

Swayamdipta, S., Thomson, S., Dyer, C., and Smith, N. A. (2017). Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *CoRR*, abs/1706.09528.

Tan, S.-S. and Na, J.-C. (2019). Positional attention-based frame identification with bert: A deep learning approach to target disambiguation and semantic frame selection. *arXiv, cs.CL*, 1910.14549.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In I. Guyon, et al., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wehrmann, J., Cerri, R., and Barros, R. (2018). Hierarchical multi-label classification networks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5075–5084. PMLR.

Xu, P. and Barbosa, D. (2018). Neural fine-grained entity type classification with hierarchy-aware loss. 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 16–25, New Orleans,

Louisiana. Association for Computational Linguistics.

Yang, B. and Mitchell, T. (2017). A joint sequential and relational model for frame-semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Copenhagen, Denmark. Association for Computational Linguistics.

Yang, C., Liu, Z., Tu, C., Shi, C., and Sun, M. (2021). *Network Embedding: Theories, Methods, and Applications*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Zhou, J., Ma, C., Long, D., Xu, G., Ding, N., Zhang, H., Xie, P., and Liu, G. (2020). Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1106–1117, Online. Association for Computational Linguistics.