Butterfly Effects in Frame Semantic Parsing: impact of data processing on model ranking

Alexandre Kabbach
Department of Linguistics
University of Geneva

Corentin Ribeyre
Etermind

Aurélie Herbelot
Center for Mind/Brain Sciences &
Dept. of Information Engineering
and Computer Science
University of Trento

{firstname.lastname}@{unige.ch;etermind.com;unitn.it}

Abstract

Knowing the state-of-the-art for a particular task is an essential component of any computational linguistics investigation. But can we be truly confident that the current state-of-the-art is indeed the best performing model? In this paper, we study the case of *frame semantic parsing*, a well-established task with multiple shared datasets. We show that in spite of all the care taken to provide a standard evaluation resource, small variations in data processing can have dramatic consequences for ranking parser performance. This leads us to propose an open-source standardized processing pipeline, which can be shared and reused for robust model comparison.

Title and Abstract in French

Effets papillon en analyse automatique de cadres sémantiques:
impact du traitement des données sur la hiérarchie des modèles

Connaître l'état-de-l'art pour une tâche donnée est un élément essentiel de toute entreprise de linguistique computationnelle. Peut-on néanmoins s'assurer que l'état-de-l'art connu est bel et bien le meilleur des modèles ? Dans ces travaux nous nous intéressons au cas de l'analyse automatique de cadres sémantiques, une tâche bien établie disposant de multiples jeux de données partagés. Nous montrons qu'en dépit du soin porté à la standardisation du processus d'évaluation, de faibles variations dans le prétraitement des données peuvent conduire à une remise en question du classement final des analyseurs. Cette constatation nous conduit à proposer une plateforme libre de traitement standardisée permettant d'obtenir des données de comparaisons fiables entre les performances des différents analyseurs.

1 Introduction

A typical contribution to Computational Linguistics research is a new *model* for a specific task, which improves performance on a known dataset. The proposed model, whilst being the object of focus in the associated publication, normally relies on some potentially extensive pre- and post-processing of data which may only be briefly reported. As shown in the seminal work by (Fokkens et al., 2013), small alterations to such processing can result in dramatic changes in model performance. In this paper, we pick up on this point and make a case for harmonizing evaluations by sharing not just datasets but entire experimental pipelines. We illustrate our point by focusing on recent frame semantic parsing literature.

Frame semantic parsing is the task of automatically extracting semantic structures in text following the theory of Frame Semantics (Fillmore, 1982) and the framework of FrameNet (Baker et al., 1998, hereafter FN). Formally defined in the SemEval 2007 shared task 19 (Baker et al., 2007), it consists of three separate subtasks: (1) *target identification*: the task of identifying all frame evoking words in a given sentence; (2) *frame identification*: the task of identifying all frames of pre-identified targets in a given sentence; and (3) *argument identification*: the task of identifying all frame-specific frame

Proceedings of the 27th International Conference on Computational Linguistics, pages 3158–3169
Santa Fe, New Mexico, USA, August 20-26, 2018.

element spans and labels for pre-identified targets in a given sentence. The following sentence provides an example of FN annotation for the predicate *splash*.v evoking the `Cause_fluidic_motion` frame, and its corresponding FLUID, SOURCE and GOAL arguments:

(1) $[\text{Wine}]_{\text{Fluid}}$ was **splashed** $[\text{from jug}]_{\text{Source}}$ $[\text{to cup}]_{\text{Goal}}$ and often drained in one loud gulp.

Thanks to the shared task, the FN community has produced valuable datasets to evaluate models against. Still, as we will show in this paper, the actual evaluation settings used by various research groups exhibit differences which can in some cases have dramatic consequences for the reported results. In particular, all available research on frame semantic parsing relies on some form of pre- and post-processing of FN XML data. Post-processing usually involves converting FN train, dev and test XML splits to standard CONLL-based formats while removing erroneous or problematic annotation. Preprocessing involves part-of-speech tagging *ad minima*, and often lemmatization and dependency parsing as well. We will see that those pre- and post-processing pipelines turn out to be rather inconsistent across published investigations (see §2.2), preventing fair comparison between models. In fact, we demonstrate that when using a cleaned up and standardized pipeline, the ranking of existing parsers can change quite substantially.

The deviations we observe across experimental setups cover various aspects of pre- and post-processing. First, inconsistencies in the post-processing pipeline lead parsers to be actually evaluated on slightly different test sets across studies. Moreover, those test sets contain duplicate sentences and annotations – leading parsers to be scored multiple times on the same data – as well as overlapping data between train and test sets – thereby artificially improving overall scores. Second, inconsistencies in the pre-processing pipeline prevent fair comparison across models by conflating the contribution of the pre-processing toolkit with that of the statistical model itself. We further note that all recent work on frame semantic parsing rely on the FN 1.5 dataset, while the FN 1.7 dataset, available since 2016, contains nearly 20% more gold annotated data, allowing for larger train and test sets and hence more robust baselines.

The goal of this paper is to highlight the impact of small deviations in experimental settings on model performance. Our contributions are fourfold: we (1) replicate past results of frame semantic parsing on a single robust pre- and post-processing pipeline; (2) quantify the impact of the pipeline on model performance; (3) analyze the robustness of past results when tested on a normalized FN 1.7 setup; and (4) provide robust baselines on both FN 1.5 and FN 1.7 experimental setups for future research.

We focus specifically on replicating frame identification results of the SIMPLEFRAMEID system of Hartmann et al. (2017), and argument identification results of the SEMAFOR system of Das et al. (2014), as modified by Kshirsagar et al. (2015), and the OPEN-SESAME system of Swayamdipta et al. (2017). We exclude the FRAMAT system of Roth and Lapata (2015) for brevity, as it uses a different evaluation script on argument identification given gold frames, and we ignore all other contributions (Hermann et al., 2014; Täckström et al., 2015; FitzGerald et al., 2015; Roth, 2016; Yang and Mitchell, 2017) which do not provide open source systems.[1]

Finally, rather than releasing a fixed preprocessed FrameNet dataset like Bauer et al. (2012), we propose an application for testing various frame semantic parsers in custom experimental setups. Our application allows converting FrameNet XML data to various standard NLP formats (such as, e.g., BIO or CoNLL-X) relying on a common architecture which handles data processing side effects observed throughout this work. Preprocessing pipelines can thereby be easily modified while guaranteeing the consistency and robustness of the experimental setups across parsers.

The structure of the paper is as follows: in Section 2, we introduce all past research on frame semantic parsing replicated throughout this work, the statistical models used, as well as their experimental and evaluation setups. In Section 3 we introduce our own experimental setup designed to overcome the limitations observed in past research. In Section 4, we provide both frame and argument identification scores for past models trained on our own experimental setup, on both FN 1.5 and 1.7 datasets. In Section 5 we discuss the impact of pre-processing on argument identification, and in Section 6 and Section 7 we provide new robust baselines on both FN 1.5 and 1.7 datasets and conclude.

[1] For (Roth, 2016), although decoding and models are open source, training software is not

2 Existing frame semantic parsing pipelines

2.1 Models

SIMPLEFRAMEID The *frame identification* model of SIMPLEFRAMEID (Hartmann et al., 2017) is based on distributed representations of predicates and their (syntactic) context. The overall model relies on a disambiguation classifier which learns weights for all frames in the lexicon, given a vector space model providing dense vector representations for the predicate and its context. The context of the predicate can include all words in the sentence (SENTBOW model), or be restricted to direct dependents of the predicate (DEPBOW model). The overall classifier can rely on two distinct classification methods: a two-layer neural network (NN) or a WSABIE-based method following the approach of Hermann et al. (2014), consisting in mapping input representations and frame representations to a common latent space using the WSABIE algorithm (Weston et al., 2011). In the standard decoding setup, the best scoring frame is selected among all possible frames of a given predicate, as specified in the lexicon. If the predicate is not found in the lexicon, the best scoring frame is selected among all frames, while if the predicate is unambiguous and has a single possible frame, the frame is assigned directly. Hartmann et al. (2017) provide three evaluation metrics for all their models: *total*, which provides an accuracy score for all predicates in the test set; *ambig* which provides accuracy scores for ambiguous predicates only (predicates with more than one possible frames in the lexicon); and *no-lex*, accuracy scores while treating each predicate as *unknown* and scoring all frames in the lexicon.

SEMAFOR The *argument identification* model of SEMAFOR is a system originally proposed by Das et al. (2014) and modified by Kshirsagar et al. (2015). The model derives the set of argument spans via a rule-based algorithms over the syntactic context of a predicate word, and labels roles using a conditional log-linear model over spans for each role of each evoked frames, trained using maximum conditional log-likelihood. The original model makes extensive use of handcrafted features, ranging from lexical items combinations to syntactic dependency paths. To the original features are added the *exemplar* and *hierarchy* features (henceforth EX and H) proposed by Kshirsagar et al. (2015). EX incorporates FN exemplar data – lexicographic examples annotating a single predicate – to the original 'fulltext' train set, while H incorporates information regarding specific frame relations (Inheritance and SubFrame). Decoding is done using beam search which produces a set of k-best hypotheses of sets of span-role pairs. The approach enforces multiple constraints including the fact that a frame element label may be assigned to at most one span and that spans of overt arguments must not overlap.

OPEN-SESAME OPEN-SESAME (Swayamdipta et al., 2017) is another *argument identification* system based on the SegRNN model of Kong et al. (2015), which relies on a combination of bidirectional RNNs and semi-Markov CRF – with a slight modification to favor recall over precision – to learn embedded representations of targets, lexical units, frames, frame elements and spans. In its basic form, the OPEN-SESAME parser is syntax-free, although it may also incorporate syntactic cues such as dependency parses or phrase structures. For brevity, and to ease the replication process, we focus in this work on the syntax-free and dependency-based models only. Note that, throughout this work, we report results with single models and not ensemble models, and that, due to differences in initialization, variations of F_1 scores can be observed over our reported results, of up to 1 point of F_1 score on FN 1.5 setups and .5 points of F_1 score on FN 1.7 setups.

2.2 Variations in existing experimental settings

SIMPLEFRAMEID and SEMAFOR rely on the original DAS sets (Das and Smith, 2011; Das et al., 2014). OPEN-SESAME extracts its test set relying on the exact same list of FN fulltext as Das and Smith (2011), but via a different pre-processing pipeline. We found both test splits to contain duplicate sentences and annotationsets,[2] as well as at least one overlapping sentence and corresponding annotationsets between train and test splits.[3] The DAS test split contains 4457 annotationsets when the OPEN-SESAME test

[2] In FN, an annotationset is defined as one set of labels for a predicate and its corresponding arguments in a given sentence.

[3] The sentence in question being *Even if Iran possesses these biological agents, it faces a significant challenge in their weaponization and delivery* .

split contains 4428 annotationsets. Corresponding parsers are therefore not evaluated on the exact same set of gold data. The difference in the number of annotationsets is partially explained by the fact that OPEN-SESAME relies on the BIO tagging scheme, which does not support overlapping frame elements, to generate its gold data. Of the 4457 annotationsets of the DAS split, at least 162 were actually duplicates. Of the 2420 sentences listed in both test sets, only 982 contained annotation, and 107 of them were duplicates, making for an actual test set of 875 unique annotated sentences.

In (Das et al., 2014; Kshirsagar et al., 2015; Hartmann et al., 2017), data are tokenized with SED, part-of-speech tagged with MXPOST (Ratnaparkhi, 1996), lemmatized with WORDNET (Miller, 1995) and dependency-parsed with the MST parser (McDonald et al., 2006). In (Swayamdipta et al., 2017), data retain the original FN tokenization, are lemmatized with NLTK (Bird et al., 2009) and part-of-speech-tagged and dependency-parsed with SYNTAXNET (Andor et al., 2016). To incorporate exemplars into the training set, Kshirsagar et al. (2015) apply an extra layer of filtering by excluding annotationsets containing no overt frame element labels from the training data. As detailed in §5.3, this has a non-negligible impact on argument identification and we treat this filtering layer as an independent feature.

All the above systems generate their *lexicons* by scanning the *entirety* of FN data. This includes the predicate-to-frame map of SIMPLEFRAMEID, the frame-to-frame-element map of SEMAFOR and OPEN-SESAME, and the frame-to-frame and frame-element-to-frame-element relation maps of SEMAFOR when used with the *hierarchy* feature. Such an approach leads systems to have partial access to information which is only included in the test set, such as the number of frames available for an unknown predicate,[4] the list of frame elements of a given unknown frame, and the relation between an unknown frame and a known frame.

2.3 Evaluation software

Throughout this work we rely on two distinct evaluation scripts, motivated by considerations over the design of the original perl evaluation script of the SemEval shared task 19 on frame structure extraction. Our first script, called SEMEVAL, is the original SemEval evaluation script as-is, and is used to score argument identification given predicted frames. Our second script, called ARGSONLY, is a modified version of the evaluation script originally provided by Kshirsagar et al. (2015), which we use to score argument identification given gold frames. Both scripts provide global precision, recall and F_1 scores microaveraged across a concatenation of the test set.

Modifications to the original SEMEVAL script are motivated by its computing scores for both frames and arguments jointly, which dilutes contributions to argument identification in gold frames settings by systematically granting extra credits for gold frames. Unlike the evaluation script of Kshirsagar et al. (2015) however, our ARGSONLY evaluation does not skip the evaluation of frame-only annotation sets with no arguments, as it would not penalize parsers for over-predicting arguments.

3 Building a new harmonized experimental setup

3.1 Datasets and post-processing

We generate our train, dev and test splits following the list of fulltext documents provided by Swayamdipta et al. (2017). For both FN 1.5 and FN 1.7 datasets, dev and test splits are generated from the same list of FN fulltexts. To guarantee no duplicates and no overlap within and across datasets, we apply a strong filter based on an annotationset hash generated from the concatenated annotationset sentence text hash and the annotationset target hash. The sentence text hash is the sentence text lower-cased on stripped of all its whitespaces while the target hash is a concatenation between the target predicate string and the target indexes. To guarantee near-perfect matching between SEMEVAL-formatted test XML data and the original gold FN XML data, we directly convert FN XML data without relying on intermediate formats (e.g. CONLL or BIO formats), and retain all annotationsets except those with inconsistent labels.[5]

[4] I.e. a predicate in the test set not seen during training.

[5] We define inconsistent labels as those with inconsistent start/end indexes, e.g. when only one of the two indexes is missing. Note that when such annotationsets are found, we filter out the entire annotationset and not just the erroneous label, as we consider the entire annotationset as an indivisible piece of information

Table 1 details the number of sentences and annotationsets in both FN 1.5 and 1.7 setups. The 50% increase in test gold data between FN 1.5 and 1.7 guarantees more robust baselines when testing on the 1.7 setup.

	TRAIN 1.5 FT	TRAIN 1.5 FT+EX	TEST 1.5	TRAIN 1.7 FT	TRAIN 1.7 FT+EX	TEST 1.7
#sent	2,654	150,426	875	3,362	168,792	1,247
#anno	16,706	170,889	4,148	19,550	192,554	6,446

Table 1: Number of sentences and annotationsets in train and test splits for both FN 1.5 and FN 1.7 setups, with training data generated from fulltexts only (FT) or fulltexts and exemplars (FT+EX)

3.2 Pre-processing

In all our experiments we retain the original FN tokenization, while lemmatizing data with NLP4J (Choi, 2016). For part-of-speech tagging, we experiment with MXPOST (Ratnaparkhi, 1996) and NLP4J, and for dependency parsing with the MST (McDonald et al., 2006) and BIST (Kiperwasser and Goldberg, 2016) parsers. The BIST parser is used in both its graph-based (BMST) and transition-based (BARCH) variants. Both MST and BIST parsers are trained on sections 02-21 of the Penn TreeBank (Marcus et al., 1993). We choose MXPOST and MST for the purpose of replicating previous studies, and NLP4J and BIST for performance. At the time when this research started, state-of-the-art results on frame semantic parsing were reported by Roth (2016) who recommended the use of both NLP4J and BIST for pre-processing.

3.3 Frame semantic parsers

We fork the branch of SEMAFOR used in (Kshirsagar et al., 2015), as well as the SIMPLEFRAMEID and OPEN-SESAME master branches. For each parser, we re-implement lexicon creation methods in order to include only data seen during training (see §3.1). That is, we reproduce the predicate-frame map of SIMPLEFRAMEID, the frame-to-frame-element map of SEMAFOR and OPEN-SESAME, and the hierarchy feature-related relation maps of SEMAFOR. We keep all parsers and hyperparameters as detailed in (Hartmann et al., 2017), (Kshirsagar et al., 2015) and (Swayamdipta et al., 2017), and set the regularization parameter λ for SEMAFOR to $\lambda = 10^{-5}$.

3.4 Pipeline release

To ease replication and future work on frame semantic parsing, we release a single Python application called PYFN, which provides a set of Python models to process FN annotation. We thereby enforce data consistency across experiments, as data conversion[6] is processed via a single set of models. We create a larger standalone bundle including all forked parsers used as well as all datasets, resources and scripts necessary to replicate our experiments, and release it open source at `https://gitlab.com/akb89/pyfn`. Each experiment introduced in this work is labeled with a unique #id, to which corresponds a README file listing all necessary instructions for replicating the experiment, found under the *experiments* directory of the PYFN repository.

4 Replication of previous studies

The purpose of this section is to measure the contribution of each model introduced in §2.1 on our robust experimental setup (§3.1). Given that we use slightly different test sets and lexicons from the original studies, we focus on comparing the order of magnitude of each contribution and the resulting ranking of models, as well as analyzing the robustness of each contribution when tested on the FN 1.7 dataset.

4.1 Frame identification: the SIMPLEFRAMEID system

On frame identification, Tables 3 and 4 confirm one of the main results of Hartmann et al. (2017) presented in Table 2, namely, that the NN model outperforms the WSABIE model, and this in both FN 1.5

[6]E.g. to and from FN XML format, SEMEVAL XML format, CONLL-09 format, CONLL-X format, BIO tagging format, etc.

and FN 1.7 setups. Table 3 also confirms the hierarchy between models on the FN 1.5 setup, albeit not as clearly as originally reported, with differences between NN and WSABIE models closer to 2 points of accuracy rather than 3. However, Table 4 shows that the hierarchy between the NN + SENTBOW and NN + DEPBOW models are not robust across datasets, as the NN + DEPBOW model performs better than the NN + SENTBOW model on the FN 1.7 setup, by a margin of .6 accuracy point overall.[7] We conclude that the superiority of the NN + SENTBOW model cannot be generically ascertained, in line with original experiments reported by Hartmann et al. (2017).

Hartmann et al. (2017) FRAMENET 1.5 SETUP					
system	pos	dep	total	ambig	no-lex
WSB + SENTBOW	MXPOST	MST	84.5	67.6	72.0
WSB + DEPBOW	MXPOST	MST	85.7	69.9	71.2
NN + SENTBOW	MXPOST	MST	**87.6**	**73.8**	**77.5**
NN + DEPBOW	MXPOST	MST	87.5	73.6	76.5

Table 2: SIMPLEFRAMEID frame identification accuracy scores on DAS splits

(OUR) FRAMENET 1.5 SETUP				
xp	system	total	ambig	no-lex
#46	WSB + SENTBOW	81.2	66.8	72.9
#46	WSB + DEPBOW	81.4	69.1	74.3
#46	**NN + SENTBOW**	**83.1**	**73.0**	77.0
#46	NN + DEPBOW	82.9	72.7	**78.3**

Table 3: SIMPLEFRAMEID frame identification accuracy scores on our **FN 1.5** splits

(OUR) FRAMENET 1.7 SETUP				
xp	system	total	ambig	no-lex
#67	WSB + SENTBOW	80.9	64.9	73.2
#67	WSB + DEPBOW	81.7	67.1	72.5
#67	NN + SENTBOW	82.4	69.8	**77.0**
#67	**NN + DEPBOW**	**83.0**	**71.7**	76.1

Table 4: SIMPLEFRAMEID frame identification accuracy scores on our **FN 1.7** splits

4.2 Argument identification: OPEN-SESAME and SEMAFOR

On argument identification, we first try and replicate past results relying on our FN 1.5 and FN 1.7 splits, as well as a common pre-processing pipeline combining MXPOST and MST, in order to match as closely as possible the pipeline most widely used in past studies (see Table 5). We report results for the standard version of SEMAFOR and for all its adaptations introduced in (Kshirsagar et al., 2015), which include the hierarchy feature (H), the exemplar feature (EX) and the filtering feature (F).

Argument identification on FN 1.5				Gold Frames			Predicted Frames		
	POS	DEP		P	R	F_1	P	R	F_1
SEMAFOR	MXPOST	MST	(Das et al., 2014)	65.6	53.8	59.1	-	-	66.8
SEMAFOR H	MXPOST	MST	(Kshirsagar et al., 2015)	67.2	54.8	60.4	-	-	-
SEMAFOR EX + F	MXPOST	MST	(Kshirsagar et al., 2015)	66.0	58.2	61.9	-	-	-
SEMAFOR H + EX + F	MXPOST	MST	(Kshirsagar et al., 2015)	66.0	60.4	63.1	-	-	67.9
OPEN-SESAME	SYNTAXNET	-	(Swayamdipta et al., 2017)	64.7	**61.2**	62.9	68.0	**68.1**	68.0
OPEN-SESAME	SYNTAXNET	SYNTAXNET	(Swayamdipta et al., 2017)	**69.4**	60.5	**64.6**	**71.0**	67.8	**69.4**

Table 5: Argument identification scores given gold and predicted frames on DAS splits, as originally reported in (Kshirsagar et al., 2015; Swayamdipta et al., 2017). Frames are predicted with the proprietary system of Hermann et al. (2014)

[7] Note that decrease in absolute score was expected given that all models use a reduced lexicon compared to (Hartmann et al., 2017), although the unpredictability of performance on duplicates in the DAS test set could have compensated either way.

Table 6 shows that, contrary to what was previously reported, the OPEN-SESAME parser does not outperform the best version of SEMAFOR on the FN 1.5 splits. Although the dependency-based version of OPEN-SESAME does achieve comparable results than SEMAFOR, the syntax-free version is outperformed by a margin of .9 points of F_1 score. The contribution of each H, EX and F feature is confirmed across datasets, with similar order of magnitudes. Ranking of models changes however across datasets, and results on the FN 1.7 splits (Table 7) provide a ranking of models closer to the one reported in Table 5.

XP	FRAMENET 1.5 SETUP			P	R	F_1
#42	SEMAFOR	MXPOST	MST	59.1	54.3	56.6
#86	SEMAFOR H	MXPOST	MST	**60.3**	54.9	57.5
#75	SEMAFOR EX + F	MXPOST	MST	59.9	58.9	59.4
#89	**SEMAFOR H + EX + F**	MXPOST	MST	60.1	**60.6**	**60.4**
#43	OPEN-SESAME	MXPOST	-	59.7	59.4	59.5
#44	OPEN-SESAME	MXPOST	MST	59.8	59.8	59.8

Table 6: Argument identification scores given gold frames on our **FN 1.5** splits while pre-processing with MXPOST and MST

XP	FRAMENET 1.7 SETUP			P	R	F_1
#54	SEMAFOR	MXPOST	MST	61.2	53.5	57.1
#61	SEMAFOR H	MXPOST	MST	62.5	53.7	57.8
#97	SEMAFOR EX + F	MXPOST	MST	**64.8**	54.9	59.4
#63	SEMAFOR H + EX + F	MXPOST	MST	63.6	57.4	60.4
#53	OPEN-SESAME	MXPOST	-	62.9	58.4	60.6
#80	OPEN-SESAME	MXPOST	MST	63.2	**58.9**	**61.0**

Table 7: Argument identification scores given gold frames on our **FN 1.7** splits while pre-processing with MXPOST and MST

So while frame identification results proved rather consistent across setups in §4.1, argument identification results proved more erratic, with a clear mismatch of model hierarchy on the FN 1.5 setup, contradicting previous studies.

5 Impact of the preprocessing pipeline

In this section we report on the impact of different preprocessing pipelines on argument identification scores for both the SEMAFOR and OPEN-SESAME systems. Note that in §5.1 and §5.2, we compare OPEN-SESAME to the standard SEMAFOR model, without the H, EX or F features. However, similar trends can be observed for the feature-expanded versions of SEMAFOR (see §5.3).

5.1 Part-of-speech tagging

Table 8 shows a systematic improvement of F_1 score on the FN 1.5 splits when using NLP4J over MXPOST for part-of-speech tagging (compare with Table 6: #69 vs #42, #78 vs #43, and #79 vs #44). Results presented in Table 9 (compare with Table 7) tend to show that increase in performances with NLP4J is not robust across datasets (see notably #56 vs #54). However, further experiments, presented in §5.2 and §5.3 show that performance on argument identification is highly dependent on the *combination* of the part-of-speech tagger *and* the dependency parser used. In our case, the combination of NLP4J with the BIST dependency parser actually outperforms any MXPOST-based pre-processing pipeline.

XP	FRAMENET 1.5 SETUP			P	R	F_1
#69	SEMAFOR	NLP4J	MST	60.2	55.4	57.7
#78	OPEN-SESAME	NLP4J	-	59.8	60.2	60.0
#79	OPEN-SESAME	NLP4J	MST	61.2	60.2	60.7

Table 8: Argument identification scores given gold frames on our **FN 1.5** splits while pre-processing with NLP4J and MST

XP	FRAMENET 1.7 SETUP			P	R	F_1
#56	SEMAFOR	NLP4J	MST	59.1	53.4	56.1
#48	OPEN-SESAME	NLP4J	-	63.1	58.8	60.9
#55	OPEN-SESAME	NLP4J	MST	64.4	59.6	61.9

Table 9: Argument identification scores given gold frames on our **FN 1.7** splits while pre-processing with NLP4J and MST

5.2 Dependency-parsing

Table 10 and Table 11, show a near-systematic improvement of F_1 score for both SEMAFOR and OPEN-SESAME systems when using the BIST dependency parser over MST. On FN 1.5 , the contribution of the dependency parser itself to overall performance proves non-negligible, with an observed improvement ranging from .6 points of F_1 score for OPEN-SESAME between MST and BARCH (#79 and #83), to 4

points of F_1 score for SEMAFOR between MST and BARCH (#69 and #81). On certain FN 1.5 setups, the combination of pos tagger and dependency parser may even account, in order of magnitude, for more than half of the previously reported contribution of the OPEN-SESAME model over the SEMAFOR model (consider, e.g., #82 in Table 10 and #44 in Table 6 compared to the 1.5 F_1 score improvement reported in Table 5 between the best SEMAFOR model and the dependency-based OPEN-SESAME model).

XP	FRAMENET 1.5 SETUP			P	R	F_1
#69	SEMAFOR	NLP4J	MST	60.2	55.4	57.7
#70	SEMAFOR	NLP4J	BMST	67.5	56.4	61.4
#81	SEMAFOR	NLP4J	BARCH	**67.6**	**56.8**	**61.7**
#79	OPEN-SESAME	NLP4J	MST	61.2	60.2	60.7
#82	OPEN-SESAME	NLP4J	BMST	**64.1**	59.6	61.2
#83	OPEN-SESAME	NLP4J	BARCH	61.4	**61.3**	**61.3**

Table 10: Argument identification scores given gold frames on our **FN 1.5** splits while pre-processing with NLP4J and MST, BMST or BARCH

XP	FRAMENET 1.7 SETUP			P	R	F_1
#56	SEMAFOR	NLP4J	MST	59.1	53.4	56.1
#47	SEMAFOR	NLP4J	BMST	**64.8**	55.7	**59.9**
#84	SEMAFOR	NLP4J	BARCH	61.2	**55.8**	58.4
#55	OPEN-SESAME	NLP4J	MST	64.4	59.6	61.9
#57	OPEN-SESAME	NLP4J	BMST	**64.6**	59.3	61.8
#85	OPEN-SESAME	NLP4J	BARCH	63.6	**60.3**	**61.9**

Table 11: Argument identification scores given gold frames on our **FN 1.7** splits while pre-processing with NLP4J and MST, BMST or BARCH

5.3 Data filtering

As previously introduced in §2.2, Kshirsagar et al. (2015) apply a filtering layer on training data by removing all annotationsets with no overt role labels from the exemplar data. Such a filter may remove valid annotationsets where all arguments of the predicate are found to be *null instantiations* (Ruppenhofer et al., 2016) – which may correspond, e.g., to intransitive predicates, lexically licensed zero anaphora or imperative and passive constructions – but can also remove incompletely annotated annotationsets, where predicates have been disambiguated (frames have been assigned to specific targets), but frame element labels have yet to be specified. Such cases are much more frequent in exemplar than in fulltext data, justifying *a priori* the filtering layer of Kshirsagar et al. (2015) to prevent negatively biasing SEMAFOR.

Table 12 and Table 13 show that the filter feature has a systematic positive effect on argument identification performance, and that this result is robust across datasets and across pre-processing pipelines. Note that, in a configuration where EX and H features are combined, the addition of the F feature leads to a .5 F_1 (compare #89 and #87) to 1.4 F_1 (compare #63 and #62) increase in performance depending on the setup. This result suggests that the filtering layer applied by Kshirsagar et al. (2015) should be considered as a standalone feature, which contribution should be clearly separated from that of the exemplar feature, in order to better quantify the contribution of the exemplar feature itself.

XP	FRAMENET 1.5 SETUP			P	R	F_1
#42	SEMAFOR	MXPOST	MST	59.1	54.3	56.6
#45	SEMAFOR EX	MXPOST	MST	**63.0**	55.4	59.0
#75	SEMAFOR EX + F	MXPOST	MST	59.9	58.9	59.4
#87	SEMAFOR H + EX	MXPOST	MST	61.7	58.2	59.9
#89	SEMAFOR H + EX + F	MXPOST	MST	60.1	**60.6**	**60.4**
#70	SEMAFOR	NLP4J	BMST	**67.5**	56.4	61.4
#92	SEMAFOR EX	NLP4J	BMST	65.0	58.9	61.8
#93	SEMAFOR EX + F	NLP4J	BMST	65.0	60.0	62.4
#94	SEMAFOR H + EX	NLP4J	BMST	64.2	61.1	62.6
#95	SEMAFOR H + EX + F	NLP4J	BMST	62.7	**63.5**	**63.1**

Table 12: Argument identification scores given gold frames on our **FN 1.5** splits for various feature-augmented versions of SEMAFOR

XP	FRAMENET 1.7 SETUP			P	R	F_1
#54	SEMAFOR	MXPOST	MST	61.2	53.5	57.1
#88	SEMAFOR EX	MXPOST	MST	61.0	55.1	57.9
#97	SEMAFOR EX + F	MXPOST	MST	**64.8**	54.9	59.4
#62	SEMAFOR H + EX	MXPOST	MST	61.9	56.3	59.0
#63	SEMAFOR H + EX + F	MXPOST	MST	63.6	**57.4**	**60.4**
#47	SEMAFOR	NLP4J	BMST	64.8	55.7	59.9
#49	SEMAFOR EX	NLP4J	BMST	63.9	57.5	60.6
#73	SEMAFOR EX + F	NLP4J	BMST	64.3	59.4	61.8
#64	SEMAFOR H + EX	NLP4J	BMST	65.8	58.6	62.0
#65	SEMAFOR H + EX + F	NLP4J	BMST	**67.7**	**59.4**	**63.3**

Table 13: Argument identification scores given gold frames on our **FN 1.7** splits for various feature-augmented versions of SEMAFOR

Overall, we showed that divergence in the pre-processing pipeline can actually account for a significant part of the difference in performance previously observed across models, enforcing the need for standard pre-processing pipelines against which models can be fairly compared.

6 New baselines for FN 1.5 and FN 1.7

Building upon results of the previous sections, we introduce new baselines for both frame and argument identification on our FN 1.5 and FN 1.7 splits, relying on an updated preprocessing pipeline combining NLP4J for lemmatization and part-of-speech tagging, and the BMST variant of the BIST parser for dependency parsing. Although not necessarily the best pipeline observed in all setups, the NLP4J + BMST combination proved to be the best compromise between performance and speed, an important factor to take into account for facilitating future replication.

XP	FRAMENET 1.5 SETUP			P	R	F_1
#70	SEMAFOR	NLP4J	BMST	**67.5**	56.4	61.4
#95	SEMAFOR H + EX + F	NLP4J	BMST	62.7	**63.5**	**63.1**
#78	OPEN-SESAME	NLP4J	-	59.8	60.2	60.0
#82	OPEN-SESAME	NLP4J	BMST	64.1	59.6	61.2

Table 14: Argument identification baseline scores given gold frames on **FN 1.5** splits

XP	FRAMENET 1.7 SETUP			P	R	F_1
#47	SEMAFOR	NLP4J	BMST	64.8	55.7	59.9
#65	SEMAFOR H + EX + F	NLP4J	BMST	**67.7**	**59.4**	**63.3**
#48	OPEN-SESAME	NLP4J	-	63.1	58.8	60.9
#57	OPEN-SESAME	NLP4J	BMST	64.6	59.3	61.8

Table 15: Argument identification baseline scores given gold frames on **FN 1.7** splits

Table 14 and Table 15 confirm, in addition to Table 6 and Table 7, that the ranking of models reported in (Swayamdipta et al., 2017) is not robust across datasets and experimental setups. In the most robust setup (FN 1.7 with NLP4J and BMST reported in Table 15) SEMAFOR still outperforms both syntax-free and dependency-based versions of OPEN-SESAME, by a margin of at least 1.5 points of F_1 score.

Surprisingly however, the ranking of models is significantly modified when testing models with predicted frames output by SIMPLEFRAMEID, and a 10 points drop in recall is observed between the feature-expanded and the standard version of SEMAFOR (Tables 16 and 17). Those results suggest that the feature-expanded version of SEMAFOR acquires, most likely via the exemplar data, representations that deviate too much from that of those found in the fulltext test set, making it unable to compensate for the wrongly predicted frames of SIMPLEFRAMEID. Such considerations stress once again the possible domain-specific nature of both fulltext and exemplar data (Das et al., 2014), considerations which are reinforced by our results on frame identification trained on exemplar (see Table 18 and Table 19) which show a systematic decrease in performance on ambiguous predicates, suggesting that ambiguous predicates observed in the exemplar training data do not beneficially contribute to predictions on ambiguous predicates in the fulltext test set.

XP	FRAMENET 1.5 SETUP			P	R	F_1
#170	SEMAFOR	NLP4J	BMST	63.4	60.2	61.8
#295	SEMAFOR H + EX + F	NLP4J	BMST	**66.8**	50.3	57.4
#178	OPEN-SESAME	NLP4J	-	63.7	**64.4**	64.1
#182	OPEN-SESAME	NLP4J	BMST	66.1	64.0	**65.0**

Table 16: Argument identification baseline scores on **FN 1.5** splits, with frames predicted by SIMPLEFRAMEID trained on the **FN 1.5** train splits

XP	FRAMENET 1.7 SETUP			P	R	F_1
#147	SEMAFOR	NLP4J	BMST	63.1	60.6	61.8
#265	SEMAFOR H + EX + F	NLP4J	BMST	**68.7**	48.5	56.9
#148	OPEN-SESAME	NLP4J	-	64.0	55.2	59.3
#157	OPEN-SESAME	NLP4J	BMST	65.7	**62.9**	**64.3**

Table 17: Argument identification baseline scores on **FN 1.7** splits, with frames predicted by SIMPLEFRAMEID trained on the **FN 1.7** train splits

xp	FN 1.5	pos	dep	total	ambig	no-lex
#66	FT	NLP4J	BMST	83.2	**73.6**	**77.8**
#90	FT + EX	NLP4J	BMST	**84.6**	69.3	75.8

Table 18: Frame identification accuracy baseline on **FN 1.5** trained on fulltext (FT) and combined fulltext and exemplar (FT+EX) data

xp	FN 1.7	pos	dep	total	ambig	no-lex
#98	FT	NLP4J	BMST	82.3	**70.0**	**76.6**
#99	FT + EX	NLP4J	BMST	**83.6**	66.7	74.3

Table 19: Frame identification accuracy baseline on **FN 1.7** trained on fulltext (FT) and combined fulltext and exemplar (FT+EX) data

7 Conclusion

In this paper, we have discussed the impact of pre- and post-processing pipelines on frame semantic parsing performance. We have shown that the ranking of existing state-of-the-art systems can be severely impacted by arguably 'small' factors outside of the main models, to the point that respective model performance cannot be fully ascertained.

Our replication study drew the following conclusions. We found that on the whole, previous frame identification results by (Hartmann et al., 2017) could be replicated. Results on argument identification, however, gave a much patchier picture. Replication on a robust experimental setup showed a change in model ranking between OPEN-SESAME and SEMAFOR. Further experiments on pre-processing demonstrated that the *combination* of a particular parts-of-speech tagger and dependency parser (NLP4J and BIST) gives best performance on the task. The contribution of the dependency parser itself may be more than previously reported: we found that half of the advantage of OPEN-SESAME over SEMAFOR may be due to this pre-processing choice, closing the gap between the two models. We also highlighted that filtering the training data has a robust and very significant effect on performance.

Building upon these results, we ran two robust baselines on both FN1.5 and FN1.7. These experiments result in a different model ranking from what has previously been reported in the literature. However, we also found that the feature-expanded version of SEMAFOR suffers a huge drop in recall when given predicted rather than gold frames. Here again, it seems vital to be able to distinguish the contribution of different processing stages on the actual model.

On the back of the replication results presented here, we feel there is a need for a standardized experimental pipeline that would allow researchers to fully test the contribution of different pre- and post-processing factors independently from their model. Following on this recommendation, we release an open source toolkit[8] including all necessary scripts and datasets to replicate our experiments in a robust setting. We hope to thus foster more fine-grained analyses of past and future results in frame semantic parsing.

Acknowledgements

The authors are grateful to Meghana Kshirsagar for her help with the SEMAFOR parser.

References

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally Normalized Transition-Based Neural Networks. *CoRR*, abs/1603.06042.

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

Collin Baker, Michael Ellsworth, and Katrin Erk. 2007. SemEval-2007 Task 19: Frame Semantic Structure Extraction. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 99–104, Prague, Czech Republic, June. Association for Computational Linguistics.

Daniel Bauer, Hagen Fürstenau, and Owen Rambow. 2012. The Dependency-Parsed FrameNet Corpus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3861–3867, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1619.

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

Jinho D. Choi. 2016. Dynamic Feature Induction: The Last Gist to the State-of-the-Art. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 271–281. Association for Computational Linguistics.

[8] Available at `https://gitlab.com/akb89/pyfn`.

Dipanjan Das and Noah A. Smith. 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, June. Association for Computational Linguistics.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame Semantic Parsing. *Computational Linguistics*, 40(1):9–56.

Charles J. Fillmore. 1982. Frame Semantics. *Linguistics in the morning calm*, pages 111–137.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic Role Labeling with Neural Network Factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 960–970, Lisbon, Portugal, September. Association for Computational Linguistics.

Antske Fokkens, Marieke van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. 2013. Offspring from Reproduction Problems: What Replication Failure Teaches Us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1691–1701, Sofia, Bulgaria, August. Association for Computational Linguistics.

Silvana Hartmann, Ilia Kuznetsov, Teresa Martin, and Iryna Gurevych. 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, April. Association for Computational Linguistics.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 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, June. Association for Computational Linguistics.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. *TACL*, 4:313–327.

Lingpeng Kong, Chris Dyer, and Noah A. Smith. 2015. Segmental Recurrent Neural Networks. *CoRR*, abs/1511.06018.

Meghana Kshirsagar, Sam Thomson, Nathan Schneider, Jaime Carbonell, Noah A. Smith, and Chris Dyer. 2015. Frame semantic Role Labeling with Heterogeneous Annotations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 218–224, Beijing, China, July. Association for Computational Linguistics.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, 19(2):313–330.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual Dependency Analysis with a Two-Stage Discriminative Parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 216–220, New York City, June. Association for Computational Linguistics.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Adwait Ratnaparkhi. 1996. A Maximum Entropy Model for Part-Of-Speech Tagging. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*. Association for Computational Linguistics, May.

Michael Roth and Mirella Lapata. 2015. Context-aware Frame-Semantic Role Labeling. *Transactions of the Association for Computational Linguistics*, 3:449–460.

Michael Roth. 2016. Improving Frame Semantic Parsing via Dependency Path Embeddings. In *Book of Abstracts of the 9th International Conference on Construction Grammar*, pages 165–167, Juiz de Fora, Brazil, October.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, Collin F. Baker, and Jan Scheffczyk. 2016. FrameNet II: Extended Theory and Practice. Technical report, ICSI, Berkeley.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. *CoRR*, abs/1706.09528.

Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Efficient Inference and Structured Learning for Semantic Role Labeling. *Transactions of the Association for Computational Linguistics*, 3:29–41.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling Up To Large Vocabulary Image Annotation. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*.

Bishan Yang and Tom Mitchell. 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 1258–1267, Copenhagen, Denmark, September. Association for Computational Linguistics.