Abstractive Text Summarization using Sentential Semantics

Luis Glaser
luis.glaser@em.uni-frankfurt.de

Abstract

We present `tldr` a python package for abstractive text summarization. `tldr` uses graph structures to represent and summarize documents. It includes a parser for Semantic Role Labeling (SRL) trained on dataset with a novel extension on PropBank annotation. These extend the base annotation schema with semantic macro roles and admonial roles in order to allieviate some of the challenges state-of-the-art SRL annotation corpora and schemas face. We furthermore describe experiments we performed on the task of abstractive text summarization using this novel schema for SRL parsing and graph-based summarization.

1 Introduction

Our approach aims to solve several challenges in SRL by providing interpretable role annotations. We establish a clear relationship between semantic roles and their realizations, and differentiate systematically between core roles (Valin et al., 2004) and modifiers. By enabling corpus-based SRL instead of relying on lexicons and complementing verb-focused roles with adnominal relations, our method addresses gaps in current SRL practices. It also generalizes over existing corpora and frame inventories for various predicate constructions (Streusle, (Schneider and Smith, 2015; Schneider, Hwang, et al., 2018), PropBank (Kingsbury and Palmer, 2002), VerbNet, FrameNet TODO CITE), creating a more comprehensive annotation schema.

To address the need for enhanced semantic parsing, our framework integrates with existing resources like AMR and UCCA, offering a minimal and transparent role inventory. This ensures compatibility with downstream tasks such as OpenIE (Angeli et al., 2015) and builds on the Universal Dependencies framework (Nivre, 2020). Furthermore we aim to maintain interoperability with PropBank-based annotations, including Abstract Meaning Representation (AMR) (Banarescu et al., 2013) and UCCA (Abend and Rappoport, 2013). By seamlessly integrating with these established systems, our approach enhances the versatility and applicability of SRL across different linguistic tasks and resources.

A key difference from traditional SRL methods is our exclusive focus on semantic roles, with explicit predicates derivable from the observed role inventory and predicate lemma. By abandoning the potentially controversial distinction between core roles and modifiers, and instead using grammatical criteria and linguistic theories of argument linking, our approach provides an alternative solution to differentiate macro- and other

semantic roles. This method, rooted in earlier PropBank work and efforts to consolidate lexical resources, offers a novel and effective solution to the challenges faced by current SRL systems, hoping to improve the quality of down-stream summarizations as well.

2 Sentential Semantics

Since Fillmore (1968), semantic roles, also known as thematic roles in the Chomskyan tradition, have been extensively developed for natural language processing applications such as natural language understanding (Banarescu et al., 2013) and machine translation (Opitz et al., 2020). A variety of resources for English have emerged, each varying in depth, design, and abstraction. Among them are FrameNet and PropBank. FrameNet, the most advanced, defines frames that group lexemes and introduces a framework-specific conception of inheritance, facilitating future inferences over SRL analyses. However, FrameNet's fine-grained roles pose challenges in machine learning applications due to data sparsity and the difficulty of annotating these nuanced roles.

In contrast, PropBank is designed primarily for corpus annotation, with a frame inventory developed alongside it. PropBank defines predicates as word senses tied to specific lexemes without hierarchical organization, unlike FrameNet. It distinguishes core arguments with numbered labels (ARG0, ARG1, etc.) and modifiers with human-readable labels. This approach, extended to various languages and parts of speech, simplifies annotation by generalizing the semantics of core arguments post-corpus annotation. Despite having only a few core argument labels, PropBank's approach allows for near-infinite role variations and conventions for labeling are not systematically applied across all arguments, which leads to disambiguity.

2.0.1 Semantic Role Labeling

SRL has been traditionally split into three subtasks, predicate detection, predicate disambiguation and argument labeling (Ouchi et al., 2018). Others have already considered doing tasks in one *step*, e.g. Fei et al. combined various LSTM based models to encode the entire SRL task in one *structure*. In general the first subtask is the *easiest*, as it is merely concerned with identifying which word in a sentence acts as a predicate that later gets arguments attached to it. Depending on the underlying encoding, this argument prediction can be a single or multi token prediction. Our novel shallow encoding will naturally have inflated quality metrics as we only require the head of a multi-token predicate to be identified.

2.1 Graph-Based Summarization

Text summarization can be split into a variety of subtasks, however the most pronounced differentiation is between abstractive text summarization and extractive text summarization. The latter tries to select the most relevant sentences from a text corpus to create the summarization. Abstractive text summarization on the other hand tries to create a more *natural* summarization which tries to cover *the gist* of a document without focusing on the exact wording. This comes with a variety of challenges: The analysis of the

underlying text has to become deeper - while extractive summarization can levy more general techniques to rank the sentences, e.g. TextRank Mallick et al., 2019; Mihalcea and Tarau, 2004, abstractive text summarization has to consider the relation of meaning between sentences within a document (J. Zhang et al., 2020). Furthermore abiguity becomes a larger problem, e.g. which nodes can be merged, as they refer to the same entity and which can not despite sharing the same surface realization. Depending on the node type, Named Entity Linking (NEL) can be used to merge entities using the wikidata knowledge graph. Furthermore, coreference solution algorithms can link pronouns to their entity or resolve anaphoras. Usage of graph structures have been widely applied to text summarization tasks. modified Textrank (Mallick et al., 2019) to consider a document as a graph of sentences to select the sentences for the output summarization. Note that this work belongs to the class of extractive text summarization. (Liu et al., 2015) explored the potential of SRL annotations for the abstractive summarization task by parsing single AMR sentences graphs, combining and summarizing them but left the generation of them to future work.

2.2 Graph to Text Generation

Graph to Text Generation or more generally Data to Text Generation has seen extensive research, especially in the later years. Bevilacqua et al. (2021) tackled both the parsing and generation task for AMR as a simple `seq2seq` task. Bai et al. (2022) reported improvements caused by pre-training on multiple graph encoding and decoding tasks to learn graph structure capabilities.

3 Deliverables

We provide a corpus annotated with our sentential extension to semantic annotations.[1] This repository includes the annotated data in `CoNLL` format and both code and documentation to reproduce it. Second we created the `python` package `tldr` that contains source code to recreate the postulated together with any metrics. `tldr` furthermore provides modules that can be used to use any summarization algorithms in downstream applications. Note that the package is by no means *feature-complete* but rather meant as way to share the current state of development with others. The code is distributed under MIT license such that industry use is encouraged as well.

3.1 tldr-package

The `tldr` package is split into five modules.

1. `srl` is concerned with training and inference of the shallow srl data
2. `summarize` can summarize sentence and document graphs with a variety of techniques.
3. `generate` serializes sentence graphs into natural language.

[1] Publicly available under MIT at `https://www.github.com/glaserL/sentential-semantics`

Alice	bought	a	new	car	last	week
ARG0	PRED	ARG1	ARG1	ARG1	ARG-TMP	ARG-TMP

Table 1: Hyperparameters for the SRL tasks.

4. `data` really only creates sentences graph objects from the srl parses.

5. `tracking` contains utilities for training but also recovering already trained models.

The three modules `srl`,`summarize` and `generate` contain the main functionality of `tldr`. Each provides a independent interface. For training purposes the `tracking` package can be used to store and recover models, we currently use `mlflow` (Chen et al., 2020) for this but other experiment tracking software can replace this module.

3.2 Graph-Based Summarization

Inspired by Khan et al. (2018) we create abstract sentence structures from the previously created semantic parses. Each predicate is represented by a (sub-)tree, with the predicate as its head and the argument spans as its children. The roots are replaced with their lemma. In a second step, using an implementation of Mallick et al. (2019) modified TextRank algorithm, each subgraph is assigned a score and then the top k predicate graphs are retained in the resulting summarization graph. This allows us to remove irrelevant information from parts of a sentence that would otherwise be considered relevant and scoring these parts individually.

3.3 Graph to Text Generation

Recent advancements in Natural Language Generation (NLG) show the promise in using Large Language Models (SSRLs) for NLG. We evaluated the usage of 2 models, `TinyLlama` (P. Zhang et al., 2024) and `Mistral 7B` (Jiang et al., 2023) that are publicly available and had acceptable requirements to hardware. The used models were previously fine-tuned on instruction datasets. We considered three factors to influence the quality of the generated output: the selection of the instruction prompt, which acts as a prefix explaining the task; the sampling technique and its hyperparameters, which determine the length and quality of the generated text; and the serialization of sentence graphs, ensuring the preservation of syntactic and semantic structures.

3.4 Serialization

We tried four techniques to serialize the data. First, the sentence graph is serialized as a dictionary similar to json, including brackets, lists etc. Second, a tabular data structure where the arguments are provided per line. The third and fourth serialization techniques abandon the structured approach and encode the annotation name as a key and the tokens as values and vice-versa, See fig. 1 for examples.

```
{
  "PRED" : "bought",
  "ARG0" : ["Alice"],
  "ARG1" : ["a", "new", "car"],
  "ARG-TMP" : ["last", "week"]
}
```

```
PRED : bought
ARG0 : Alice
ARG1 : a new car
ARG-TMP : last week
```

```
{
  "PRED" : "bought",
  "ARG0" : "Alice",
  "ARG1" : "a new car",
  "ARG-TMP" : "last week"
}
```

```
bought: PRED
Alice: ARG0
a new car: ARG1
last week: ARG-TMP
```

Figure 1: Multiple possible serializations for the sentence in table 1.

4 Experiments

Changing the underlying data annotation format in addition to a novel summarization technique requires thorough validation whether the changes have a positive effect on the quality of the outcome. We therefore undertook various experiments.

4.0.1 Semantic Parsing

In order to harness the novel dataset, a parser which is fine-tuned on the data is required. We decided against fine-tuning an existing SRL parser for two reasons. First, in order to control for the effect of the change in data annotations. Second to avoid polluting our data set as they might have been trained on the base dataset already.

As previously outlined, using the extended PropBank-based annotation scheme reduces partial tasks to merely the *predicate prediction* and *argument labeling* tasks.

To correctly measure the impact on both tasks we fine-tuned a model for each task. We used the PMB corpus to derive the shallow annotated version.

4.0.2 Graph to Text Generation

Firstly we controlled the summarization technique by not performing any summarization and merely regenerating the original sentence in the corpus. We evaluated the impact of each serialization technique using BLEU score (Papineni et al., 2001). For this we parsed the validation split of the CNN corpus (Hermann et al., 2015) using our novel SRL parser. We then regenerated the original text.

This answers two questions: Is using sentences graphs as an intermediate format to perform linguistic modifcations on a valid technique, respectively how well does it perform? Secondly, what serialization and sampling methods should be used in order to achieve the best performance?

4.0.3 Summarization

The final experiment was performed for the central task, the text summarization itself. Again using the CNN Corpus(Hermann et al., 2015) we used the best performing SRL

Corpus Name	Number of Tokens	Number of unique predicates
PMB	$1,254,613$	1418

Table 2: Number of tokens for each corpus in the sentential semantics collection.

Parameter Name	predicate	arguments
F1 (eval)	0.98	0.84
Loss (eval)	0.14	0.42
Loss (total)	0.0038	0.088
Loss (train)	0.022	0.18

Table 3: Best performing model for the SRL tasks.

parser to annotate the CNN corpus. We then did the summarization. Note that in order to avoid data pollution issues we only did this using the test set split, not the entire set.

The final summarization was evaluated using the ROUGE score (Lin, 2004).

5 Results

5.1 Sentential Semantics Dataset

One central contribution of this work are the sentential semantics corpora[2] that are the basis for the SRL parser. Table 2 shows the number of tokens in each base corpus contained in our corpus collection.

5.2 Semantic Parsing

3 shows a variety of metrics for both predicate prediction and argument labeling tasks. The predicate prediction task performed well, with F1 of 0.98. The argument span labeling task performed still fairly well with F1 of 0.84. 3 shows the hyperparameters used to create this best-performing model - it's worth to highlight that the base model is DistilBERT Sanh et al., 2019 , which promises to make further fine-tuning more time and energy efficient. As a side note, the exact same training on the argument prediction task took 12 hours on the *standard* BERT Devlin et al., 2019 model while the DistilBERT version took only 4.5 hours. Technically the BERT model outperformed the DistilBERT model (eval F1 of 0.847 compared to eval F1 0.843) which we argue is not worth the performance hit, thus selecting the technically 'worse' model as champion.

5.3 Graph to Text Generation

Recreating the original sentences from the silver annotated produced a top oerformance of 0.4 BLEU score[3]. It used contrastive search (Su et al., 2022) with $k = 57$, $p = 0.6$,

[2]Publically available at `https://www.github.com/glaserL/sentential-semantics`.

[3]Precision 0: 0.61, 1: 0.44, 2: 0.35, 3 0.28

and the `Mistral 7B` model (Jiang et al., 2023). The best performing prompt was *Generate a sentence from the following semantic parse:*.

The temperature was 0.6, which intuitively a low temperature does make sense as the generation should be as close to the input data - the task at hand is not creative writing.

5.4 Experiment Setup

Hyperparameter optimization was done using the `optuna` package Akiba et al., 2019. We used CNN Dataset (Hermann et al., 2015) to evaluate the performance of the novel approach. The `rogueLsum` metric only reached 0.3, which hints to major issues in the summarization technique. In future work, the summarization algorithm should be selected more carefully.

6 Discussion

Our approach offers practical benefits by using the extended sentential semantic representations instead of PropBank or FrameNet, mainly because it avoids common issues in machine-learning-driven NLP. Unlike PropBank's inconsistent numerical labels, our method directly encodes semantic roles, enabling more robust generalizations. Compared to FrameNet, our role inventory is smaller, reducing sparsity issues, and unlike PropBank, it doesn't require predicate disambiguation. Our approach also allows easier generalization to out-of-vocabulary predicates by predicting core roles from surface syntax. However, this method involves some information loss and limitations, such as treating prepositional objects like modifiers and using lemmas instead of explicit predicates. Despite these drawbacks, our method aligns with efforts like SemLink and UCCA but differs in its syntactically based role definitions and focus on SRL. However, the sentence summarization was done in a fairly naïve approach - cross sentence boundary relations are not considered in this approach. If meaning structures independent from sentence boundaries, read *below* them can be used, cross-sentence boundary merging can be considered as well. This showed in our summarization results. The parsing worked effectively in any case, allowing us to extend on it in the future.

References

Abend, Omri and Ari Rappoport (2013). "Universal Conceptual Cognitive Annotation (UCCA)". en. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria: Association for Computational Linguistics.

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019). "Optuna: a next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining*.

Angeli, Gabor, Melvin Jose Johnson Premkumar, and Christopher D. Manning (2015). "Leveraging Linguistic Structure For Open Domain Information Extraction". en. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 344–354.

Bai, Xuefeng, Yulong Chen, and Yue Zhang (2022). *Graph Pre-training for AMR Parsing and Generation*. en. arXiv:2203.07836 [cs]. URL: http://arxiv.org/abs/2203.07836 (visited on 03/15/2024).

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider (2013). "Abstract Meaning Representation for Sembanking". en. In.

Bevilacqua, Michele, Rexhina Blloshmi, and Roberto Navigli (2021). "One SPRING to Rule Them Both: Symmetric AMR Semantic Parsing and Generation without a Complex Pipeline". en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.14, pp. 12564–12573.

Chen, Andrew et al. (2020). "Developments in MLflow: A System to Accelerate the Machine Learning Lifecycle". en. In: *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning*. Portland OR USA: ACM, pp. 1–4.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". en. In: *Proceedings of the 2019 Conference of the North*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186.

Fei, Hao, Meishan Zhang, Bobo Li, and Donghong Ji (2021). "End-to-end Semantic Role Labeling with Neural Transition-based Model". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.14, pp. 12803–12811.

Fillmore, Charles J (1968). "The Case for Case". In: *Universals in linguistic theory*. New York, NY: Holt, Rinehart, and Winston, pp. 1–88.

Hermann, Karl Moritz, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom (2015). "Teaching machines to read and comprehend". In: *Proceedings of the 28th international conference on neural information processing systems - volume 1*. NIPS'15. Number of pages: 9 Place: Montreal, Canada. Cambridge, MA, USA: MIT Press, pp. 1693–1701.

Jiang, Albert Q. et al. (2023). *Mistral 7B*. arXiv:2310.06825 [cs]. URL: http://arxiv.org/abs/2310.06825 (visited on 05/31/2024).

Khan, Atif, Naomie Salim, Haleem Farman, Murad Khan, Bilal Jan, Awais Ahmad, Imran Ahmed, and Anand Paul (2018). "Abstractive Text Summarization based on Improved Semantic Graph Approach". en. In: *International Journal of Parallel Programming* 46.5, pp. 992–1016.

Kingsbury, Paul and Martha Palmer (2002). "From TreeBank to PropBank". en. In: *Proceedings of the 3rd International Conference on Language Resources and Evaluation*. Las Palmas, Canary Islands - Spain: European Language Resources Association (ELRA), pp. 1989–1993.

Lin, Chin-Yew (2004). "ROUGE: A Package for Automatic Evaluation of Summaries". en. In: Barcelona, Spain: Association for Computational Linguistics, pp. 74–81.

Liu, Fei, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith (2015). "Toward Abstractive Summarization Using Semantic Representations". en. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 1077–1086.

Mallick, Chirantana, Ajit Kumar Das, Madhurima Dutta, Asit Kumar Das, and Apurba Sarkar (2019). "Graph-based text summarization using modified TextRank". In: *Soft computing in data analytics*. Singapore: Springer Singapore, pp. 137–146.

Mihalcea, Rada and Paul Tarau (2004). "TextRank: Bringing order into text". In: *Proceedings of the 2004 conference on empirical methods in natural language processing*. Barcelona, Spain: Association for Computational Linguistics, pp. 404–411.

Nivre, Joakim (2020). "Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection". en. In: *Proceedings of the 12th Conference on Language Resources and Evaluation*. Marseille: European Language Resources Association, pp. 4034–4043.

Opitz, Juri, Letitia Parcalabescu, and Anette Frank (2020). "AMR Similarity Metrics from Principles". en. In: *Transactions of the Association for Computational Linguistics* 8, pp. 522–538.

Ouchi, Hiroki, Hiroyuki Shindo, and Yuji Matsumoto (2018). *A Span Selection Model for Semantic Role Labeling*. en. arXiv:1810.02245 [cs]. URL: http://arxiv.org/abs/1810.02245 (visited on 05/29/2024).

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2001). "BLEU: a method for automatic evaluation of machine translation". en. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*. Philadelphia, Pennsylvania: Association for Computational Linguistics, p. 311.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. Version Number: 4. URL: https://arxiv.org/abs/1910.01108 (visited on 05/28/2024).

Schneider, Nathan, Jena D. Hwang, Vivek Srikumar, Jakob Prange, Austin Blodgett, Sarah R. Moeller, Aviram Stern, Adi Bitan, and Omri Abend (2018). "Comprehensive Supersense Disambiguation of English Prepositions and Possessives". en. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 185–196.

Schneider, Nathan and Noah A. Smith (2015). "A Corpus and Model Integrating Multiword Expressions and Supersenses". en. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 1537–1547.

Su, Yixuan, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier (2022). *A Contrastive Framework for Neural Text Generation*. Version Number: 3. URL: https://arxiv.org/abs/2202.06417 (visited on 05/31/2024).

Valin, R.D. van, Rolf Kailuweit, and Martin Hummel (2004). "Semantic Macroroles in Role and Reference Grammar". en. In: *Semantische Rollen*. Tübingen: Gunter Narr Verlag, pp. 62–82.

Zhang, Jingqing, Yao Zhao, Mohammad Saleh, and Peter J Liu (2020). "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization". en. In.

Zhang, Peiyuan, Guangtao Zeng, Tianduo Wang, and Wei Lu (2024). *TinyLlama: An Open-Source Small Language Model*. arXiv:2401.02385 [cs]. URL: http://arxiv.org/abs/2401.02385 (visited on 05/31/2024).