Unsupervised Semantic Frame Induction using Triclustering

Dmitry Ustalov†, Alexander Panchenko‡, Andrei Kutuzov⋆,
Chris Biemann‡, and Simone Paolo Ponzetto†

†University of Mannheim, Germany
{dmitry,simone}@informatik.uni-mannheim.de
*University of Oslo, Norway
andreku@ifi.uio.no
‡University of Hamburg, Germany
{panchenko,biemann}@informatik.uni-hamburg.de

Abstract

We use dependency triples automatically extracted from a Web-scale corpus to perform unsupervised semantic frame induction. We cast the frame induction problem as a triclustering problem that is a generalization of clustering for triadic data. Our replicable benchmarks demonstrate that the proposed graph-based approach, Triframes, shows state-of-the-art results on this task on a FrameNet-derived dataset and performing on par with competitive methods on a verb class clustering task.

1 Introduction

Recent years have seen much work on Frame Semantics (Fillmore, 1982), enabled by the availability of a large set of frame definitions, as well as a manually annotated text corpus provided by the FrameNet project (Baker et al., 1998). FrameNet data enabled the development of wide-coverage frame parsers using supervised learning (Gildea and Jurafsky, 2002; Erk and Padó, 2006; Das et al., 2014, inter alia), as well as its application to a wide range of tasks, ranging from answer extraction in Question Answering (Shen and Lapata, 2007) and Textual Entailment (Burchardt et al., 2009; Ben Aharon et al., 2010).

However, frame-semantic resources are arguably expensive and time-consuming to build due to difficulties in defining the frames, their granularity and domain, as well as the complexity of the construction and annotation tasks requiring expertise in the underlying knowledge. Consequently, such resources exist only for a few languages (Boas, 2009) and even English is lacking domain-specific frame-based resources. Possible inroads are cross-lingual semantic annotation transfer (Padó and Lapata, 2009; Hartmann et al., 2016) or linking FrameNet to other lexical-semantic or ontological resources (Narayanan et al., 2003; Tonelli and Pighin, 2009; Laparra and Rigau, 2010; Gurevych et al., 2012, inter alia).

But while the arguably simpler task of PropBank-based Semantic Role Labeling has been successfully addressed by unsupervised approaches (Lang and Lapata, 2010; Titov and Klementiev, 2011), fully unsupervised frame-based semantic annotation exhibits far more challenges, starting with the preliminary step of automatically inducing a set of semantic frame definitions that would drive a subsequent text annotation. In this work, we aim at overcoming these issues by automatizing the process of FrameNet construction through unsupervised frame induction techniques.

Triclustering. In this work, we cast the frame induction problem as a triclustering task (Zhao and Zaki, 2005; Ignatov et al., 2015), namely a generalization of standard clustering and biclustering (Cheng and Church, 2000), aiming at simultaneously clustering objects along three dimensions (cf. Table 1). First, using triclustering allows to avoid sequential nature of frame induction approaches, e.g. (Kawahara et al., 2014), where two independent clusterings are needed. Second, benchmarking frame induction as triclustering against other methods on dependency triples allows to abstract away the evaluation of the frame induction algorithm from other factors, e.g., the input corpus or pre-processing steps, thus allowing a fair comparison of different induction models.

<table>
<thead>
<tr>
<th>FrameNet</th>
<th>Role</th>
<th>Lexical Units (LU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perpetrator</td>
<td>Subject</td>
<td>kidnapper, alien, militant</td>
</tr>
<tr>
<td>PEE</td>
<td>Verb</td>
<td>snatch, kidnap, abduct</td>
</tr>
<tr>
<td>Victim</td>
<td>Object</td>
<td>son, people, soldier, child</td>
</tr>
</tbody>
</table>

Table 1: Example of a LU tricluster corresponding to the “Kidnapping” frame from FrameNet.
The contributions of this paper are three-fold: (1) we are the first to apply triclustering algorithms for unsupervised frame induction, (2) we propose a new approach to triclustering, achieving state-of-the-art performance on the frame induction task, (3) we propose a new method for the evaluation of frame induction enabling straightforward comparison of approaches. In this paper, we focus on the simplest setup with subject-verb-object (SVO) triples and two roles, but our evaluation framework can be extended to more roles.

In contrast to the recent approaches like the one by Jauhar and Hovy (2017), our approach induces semantic frames without any supervision, yet capturing only two core roles: the subject and the object of a frame triggered by verbal predicates. Note that it is not generally correct to expect that the SVO triples obtained by a dependency parser are necessarily the core arguments of a predicate. Such roles can be implicit, i.e., unexpressed in a given context (Schenk and Chiarcos, 2016). Keeping this limitation in mind, we assume that the triples obtained from a Web-scale corpus cover most core arguments sufficiently.

Related Work. LDA-Frames (Materna, 2012, 2013) is an approach to inducing semantic frames using LDA (Blei et al., 2003) for generating semantic frames and their respective frame-specific semantic roles at the same time. The authors evaluated their approach against the CPA corpus (Hanks and Pustejovsky, 2005). ProFinder (Cheung et al., 2013) is another generative approach that also models both frames and roles as latent topics. The evaluation was performed on the in-domain information extraction task MUC-4 (Sundheim, 1992) and on the text summarization task TAC-2010. Modi et al. (2012) build on top of an unsupervised semantic role labeling model (Titov and Klementiev, 2012). The raw text of sentences from the FrameNet data is used for training. The FrameNet gold annotations are then used to evaluate the labeling of the obtained frames and roles, effectively clustering instances known during induction. Kawahara et al. (2014) harvest a huge collection of verbal predicates along with their argument instances and then apply the Chinese Restaurant Process clustering algorithm to group predicates with similar arguments. The approach was evaluated on the verb annotations.

A major issue with unsupervised frame induction task is that these and some other related approaches, e.g., (O’Connor, 2013), were all evaluated in completely different incomparable settings, and used different input corpora. In this paper, we propose a methodology to resolve this issue.

2 The Triframes Algorithm

Our approach to frame induction relies on graph clustering. We focused on a simple setup using two roles and the SVO triples, arguing that it still can be useful, as frame roles are primarily expressed by subjects and objects, giving rise to semantic structures extracted in an unsupervised way with high coverage.

Input Data. As the input data, we use SVO triples extracted by a dependency parser. According to our statistics on the dependency-parsed FrameNet corpus of over 150 thousand sentences (Bauer et al., 2012), the SUBJ and OBJ relationships are the two most common shortest paths between frame evoking elements (FEEs) and their roles, accounting for 13.5 % of instances of a heavy-tail distribution of over 11 thousand different paths that occur three times or more in the FrameNet data. While this might seem a simplification that does not cover prepositional phrases and frames filling the roles of other frames in a nested fashion, we argue that the overall frame inventory can be induced on the basis of this restricted set of constructions, leaving other paths and more complex instances for further work.

The Method. Our method constructs embeddings for SVO triples to reduce the frame induction problem to a simpler graph clustering problem. Given the vocabulary $V$, a $d$-dimensional word embedding model $v \in V \rightarrow \vec{v} \in \mathbb{R}^d$, and a set of SVO triples $T \subseteq V^3$ extracted from a syntactically analyzed corpus, we construct the triple similarity graph $G$. Clustering of $G$ yields sets of triples corresponding to the instances of the semantic frames, thereby clustering frame-evoking predicates and roles simultaneously.

We obtain dense representations of the triples $T$ by concatenating the word vectors corresponding to the elements of each triple by transforming a triple $t = (s, p, o) \in T$ into the $(3d)$-dimensional vector $\vec{t} = \vec{s} \oplus \vec{p} \oplus \vec{o}$. Subsequently, we use the triple embeddings to generate the undirected graph
Algorithm 1 Triframes frame induction

Input: an embedding model $v \in V \rightarrow \vec{v} \in \mathbb{R}^d$, a set of SVO triples $T \subseteq V^3$, the number of nearest neighbors $k \in \mathbb{N}$, a graph clustering algorithm CLUSTER.

Output: a set of triframes $F$.

1: $S \leftarrow \{t \rightarrow \vec{t} \in \mathbb{R}^{3d} : t \in T\}$
2: $E \leftarrow \{(t, t') \in T^2 : t' \in \text{NN}_{T}^{k}(\vec{t}), t \neq t'\}$
3: $F \leftarrow \emptyset$
4: for all $C \in \text{CLUSTER}(T, E)$ do
5: \hspace{1em} $f_s \leftarrow \{s \in V : (s, v, o) \in C\}$
6: \hspace{1em} $f_v \leftarrow \{v \in V : (s, v, o) \in C\}$
7: \hspace{1em} $f_o \leftarrow \{o \in V : (s, v, o) \in C\}$
8: \hspace{1em} $F \leftarrow F \cup \{(f_s, f_v, f_o)\}$
9: return $F$

$\mathcal{G} = (T, E)$ by constructing the edge set $E \subseteq T^2$. For that, we compute $k \in \mathbb{N}$ nearest neighbors of each triple vector $\vec{t} \in \mathbb{R}^{3d}$ and establish cosine similarity-weighted edges between the corresponding triples.

Then, we assume that the triples representing similar contexts appear in similar roles, which is explicitly encoded by the concatenation of the corresponding vectors of the words constituting the triple. We use graph clustering of $\mathcal{G}$ to retrieve communities of similar triples forming frame clusters; a clustering algorithm is a function $\text{CLUSTER} : (T, E) \rightarrow \mathcal{C}$ such that $T = \bigcup_{C \in \mathcal{C}} C$. Finally, for each cluster $C \in \mathcal{C}$, we aggregate the subjects, the verbs, and the objects of the contained triples into separate sets. As the result, each cluster is transformed into a triframe, which is a triple that is composed of the subjects $f_s \subseteq V$, the verbs $f_v \subseteq V$, and the objects $f_o \subseteq V$.

Our frame induction approach outputs a set of triframes $F$ as presented in Algorithm 1. The hyper-parameters of the algorithm are the number of nearest neighbors for establishing edges ($k$) and the graph clustering algorithm $\text{CLUSTER}$. During the concatenation of the vectors for words forming triples, the $|T| \times 3d$-dimensional vector space $S$ is created. Thus, given the triple $t \in T$, we denote the $k$ nearest neighbors extraction procedure of its concatenated embedding from $S$ as $\text{NN}_{T}^{k}(\vec{t}) \subseteq T$. We used $k = 10$ nearest neighbors per triple.

To cluster the nearest neighbor graph of SVO triples $\mathcal{G}$, we use the WATSET fuzzy graph clustering algorithm (Ustalov et al., 2017). It treats the vertices $T$ of the input graph $\mathcal{G}$ as the SVO triples, induces their senses, and constructs an intermedi-ate sense-aware representation that is clustered using the Chinese Whispers (CW) hard clustering algorithm (Biemann, 2006). We chose WATSET due to its performance on the related synset induction task, its fuzzy nature, and the ability to find the number of frames automatically.

3 Evaluation

Input Corpus. In our evaluation, we use triple frequencies from the DepCC dataset (Panchenko et al., 2018), which is a dependency-parsed version of the Common Crawl corpus, and the standard 300-dimensional word embeddings model trained on the Google News corpus (Mikolov et al., 2013). All evaluated algorithms are executed on the same set of triples, eliminating variations due to different corpora or pre-processing.

Datasets. We cast the complex multi-stage frame induction task as a straightforward triple clustering task. We constructed a gold standard set of triclusters, each corresponding to a FrameNet frame, similarly to the one illustrated in Table 1. To construct the evaluation dataset, we extracted frame annotations from the over 150 thousand sentences from the FrameNet 1.7 (Baker et al., 1998). Each sentence contains data about the frame, FEE, and its arguments, which were used to generate triples in the form $(\text{word}_i : \text{role}_1, \text{word}_j : \text{FEE}, \text{word}_k : \text{role}_2)$, where $\text{word}_{i/j/k}$ correspond to the roles and FEE in the sentence. We omitted roles expressed by multiple words as we use dependency parses, where one node represents a single word only.

For the sentences where more than two roles are present, all possible triples were generated. Sentences with less than two roles were omitted. Finally, for each frame, we selected only two roles, which are most frequently co-occurring in the FrameNet annotated texts. This has left us with about 100 thousand instances for the evaluation. For the evaluation purposes, we operate on the intersection of triples from DepCC and FrameNet. Experimenting on the full set of DepCC triples is only possible for several methods that scale well (WATSET, CW, k-means), but is prohibitively expensive for other methods (LDA-Frames, NOAC).

In addition to the frame induction evaluation, where subjects, objects, and verbs are evaluated together, we also used a dataset of polysemous verb classes introduced in (Korhonen et al., 2003) and employed by Kawahara et al. (2014). Statis-
The triple on a set of tuples. Consider for instance a gold sure which operates not on triples, but instead "nmPU = \text{FEE}". We use fuzzy clustering evaluation measures. Following the approach for verb class evaluation by Kawahara et al. (2014), we employ normalized modified purity (nmPU) and normalized inverse purity (niPU) as the clustering quality measures. Given the set of the obtained clusters \( K \) and the set of the gold clusters \( G \), normalized modified purity quantifies the clustering precision as the average of the weighted overlap \( \delta_{K_i}(K_i \cap G_j) \) between each cluster \( K_i \in K \) and the gold cluster \( G_j \in G \) that maximizes the overlap with \( K_i \):

\[
nmPU = \frac{1}{|K|} \sum_{i \in K} \max_{j \leq |G|} \delta_{K_i}(K_i \cap G_j),
\]

where the weighted overlap is the sum of the weights \( c_{iv} \) for each word \( v \) in \( i \)-th cluster:

\[
\delta_{K_i}(K_i \cap G_j) = \sum_{v \in K_i \cap G_j} c_{iv}.
\]

Note that nmPU counts all the singleton clusters as wrong. Similarly, normalized inverse purity (collocation) quantifies the clustering recall:

\[
niPU = \frac{1}{\sum_{j \leq |G|} \max_{i \leq |K|} \delta_{G_j}(K_i \cap G_j)}.
\]

nmPU and niPU are combined together as the harmonic mean to yield the overall clustering F-score (F₁), which we use to rank the approaches.

Our framework can be extended to evaluation of more than two roles by generating more roles per frame. Currently, given a set of gold triples generated from the FrameNet, each triple element has a role, e.g., “Victim”, “Predator”, and “FEE”. We use fuzzy clustering evaluation measure which operates not on triples, but instead on a set of tuples. Consider for instance a gold triple: (Freddy: Predator, kidnap: FEE, kid: Victim). It will be converted to three pairs (Freddy: Predator, kidnap: FEE), (kid, Victim). Each cluster in both \( K \) and \( G \) is transformed into a union of all constituent typed pairs. The quality measures are finally calculated between these two sets of tuples, \( K \), and \( G \). Note that one can easily pull in more than two core roles by adding to this gold standard set of tuples other roles of the frame, e.g., (forest, Location). In our experiments, we focused on two main roles as our contribution is related to the application of triclustering methods. However, if more advanced methods of clustering are used, yielding clusters of arbitrary modality (n-clustering), one could also use our evaluation schema.

### Baselines

We compare our method to several available state-of-the-art baselines applicable to our dataset of triples.

**LDA-Frames** by Materna (2012, 2013) is a frame induction method based on topic modeling. We ran 500 iterations of the model with the default parameters. **Higher-Order Skip-Gram (HOSG)** by Cotterell et al. (2017) generalizes the Skip-Gram model (Mikolov et al., 2013) by extending it from word-context co-occurrence matrices to tensors factorized with a polyadic decomposition. In our case, this tensor consisted of SVO triple counts. We trained three vector arrays (for subjects, verbs and objects) on the 108,073 SVO triples from the FrameNet corpus, using the implementation by the authors. Training was performed with 5 negative samples, 300-dimensional vectors, and 10 epochs. We constructed an embedding of a triple by concatenating embeddings for subjects, verbs, and objects, and clustered them using k-means with the number of clusters set to 10,000 (this value provided the best performance).

**NOAC** (Egurnov et al., 2017) is an extension of the Object Attribute Condition (OAC) triclustering algorithm (Ignatov et al., 2015) to numerically weighted triples. This incremental algorithm searches for dense regions in triadic data. A minimum density of 0.25 led to the best results. In the Triadic baselines, independent word embeddings of subject, object, and verb are concatenated and then clustered using a hard clustering algorithm: k-means, spectral clustering, or CW.

We tested various hyper-parameters of each of these algorithms and report the best results overall per clustering algorithm. Two trivial baselines are **Singletons** that creates a single cluster per instance and **Whole** that creates one cluster for all elements.

### 4 Results

We perform two experiments to evaluate our approach: (1) a frame induction experiment on the FrameNet annotated corpus by Bauer et al. (2012); (2) the polysemous verb clustering experiment on the dataset by Korhonen et al. (2003). The first is based on the newly introduced frame induction evaluation schema (cf. Section 3). The second one evaluates the quality of verb clusters only on a standard dataset from prior work.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># instances</th>
<th># unique</th>
<th># clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrameNet Triples</td>
<td>99,744</td>
<td>94,170</td>
<td>383</td>
</tr>
<tr>
<td>Poly: Verb Classes</td>
<td>246</td>
<td>110</td>
<td>62</td>
</tr>
</tbody>
</table>

Table 2: Statistics of the evaluation datasets.
Frame Induction Experiment. In Table 3 and Figure 1, the results of the experiment are presented. Triframes based on Watset clustering outperformed the other methods on both Verb F1 and overall Frame F1. The HOSG-based clustering proved to be the most competitive baseline, yielding decent scores according to all four measures. The NOAC approach captured the frame grouping of slot fillers well but failed to establish good verb clusters. Note that NOAC and HOSG use only the graph of syntactic triples and do not rely on pre-trained word embeddings. This suggests a high complementarity of signals based on distributional similarity and global structure of the triple graph. Finally, the simpler Triadic baselines relying on hard clustering algorithms showed low performance, similar to that of LDA-Frames, justifying the more elaborate Watset method.

While triples are intuitively less ambiguous than words, still some frequent and generic triples like (she, make, it) can act as hubs in the graph, making it difficult to split it into semantically plausible clusters. The poor results of the Chinese Whispers hard clustering algorithm illustrate this. Since the hubs are ambiguous, i.e., can belong to multiple clusters, the use of the Watset fuzzy clustering algorithm that splits the hubs by disambiguating them leads to the best results (see Table 3).

Verb Clustering Experiment. Table 4 presents results on the second dataset for the best models identified on the first dataset. The LDA-Frames yielded the best results with our approach performing comparably in terms of the F1-score. We attribute the low performance of the Triframes method based on CW clustering to its hard partitioning output, whereas the evaluation dataset contains fuzzy clusters. Different rankings also suggest that frame induction cannot simply be treated as a verb clustering and requires a separate task.

5 Conclusion

In this paper, we presented the first application of triclustering for unsupervised frame induction. We designed a dataset based on the FrameNet and SVO triples to enable fair corpus-independent evaluations of frame induction algorithms. We tested several triclustering methods as the baselines and proposed a new graph-based triclustering algorithm that yields state-of-the-art results. A promising direction for future work is using the induced frames in applications, such as Information Extraction and Question Answering.

Additional illustrations and examples of extracted frames are available in the supplementary materials. The source code and the data are available online under a permissive license.2

---

1 https://github.com/uhh-lt/triframes

2 https://github.com/uhh-lt/triframes
Acknowledgments

We acknowledge the support of DFG under the “JOIN-T” and “ACQuA” projects and thank three anonymous reviewers for their helpful comments. Furthermore, we thank Dmitry Egurnov, Dmitry Ignatov, and Dmitry Gnatyshak for help in operating the NOAC method using the multimodal clustering toolbox. Besides, we are grateful to Ryan Cotterell and Adam Poliak for a discussion and an implementation of the HOSG method. Finally, we thank Bonaventura Coppola for discussions and preliminary work on graph-based frame induction.

References


