Applied Network Science

RESEARCH Open Access

Semantic frame induction through the detection of communities of verbs and their arguments

Eugénio Ribeiro[1,2], Andreia Sofia Teixeira[1,3,4*], Ricardo Ribeiro[1,5] and David Martins de Matos[1,2]

*Correspondence: anmont@iu.edu
[1] INESC-ID, Lisboa, Portugal
[3] Center for Social and Biomedical Complexity, School of Informatics, Computing, & Engineering, Indiana University, Bloomington, Indiana, USA
Full list of author information is available at the end of the article

Abstract
Resources such as FrameNet, which provide sets of semantic frame definitions and annotated textual data that maps into the evoked frames, are important for several NLP tasks. However, they are expensive to build and, consequently, are unavailable for many languages and domains. Thus, approaches able to induce semantic frames in an unsupervised manner are highly valuable. In this paper we approach that task from a network perspective as a community detection problem that targets the identification of groups of verb instances that evoke the same semantic frame and verb arguments that play the same semantic role. To do so, we apply a graph-clustering algorithm to a graph with contextualized representations of verb instances or arguments as nodes connected by edges if the distance between them is below a threshold that defines the granularity of the induced frames. By applying this approach to the benchmark dataset defined in the context of SemEval 2019, we outperformed all of the previous approaches to the task, achieving the current state-of-the-art performance.

Keywords: Semantic frames, Semantic roles, Contextualized representations, Community detection, Graph clustering

Introduction

A word may have different senses depending on the context in which it appears. Conversely, different words that appear in the same context are typically related in some manner. Fillmore's theory of frame semantics (Fillmore 1976) states that these contexts, which are based on recurring experiences, can be represented in the form of semantic frames. A semantic frame is defined as a coherent structure of related concepts, such that without knowledge of all of them, one does not have complete knowledge of any of them. Using less abstract terms and partially relying on Minsky's definition in the context of knowledge representation and artificial intelligence (Minsky 1974), a semantic frame is a conceptual structure that describes a situation or entity, as well as its participants or properties. These participants are typically associated with the roles that they play in the context of the frame. Semantic roles that are specific to the frame are called frame slots or elements. Additionally, from a linguistic perspective, the participants can also be associated with generic semantic roles that are not specific to the frame, but still describe the roles played in the context of the event represented by the semantic frame. As an example, consider the sentence *"Mary sold a car to John"*. We can say that this sentence, and especially the verb *sell*, evokes the *commercial transaction* frame. Furthermore, we can identify three participants – *Mary*, *John*, and *a car* – which fill the *seller*, *buyer*, and *goods* frame slots and play the *agent*, *recipient*, and *theme* semantic roles, respectively.

Considering that semantic frames are able to represent different contexts, which can be used to disambiguate word senses and identify words that are related, sets of frame definitions and annotated datasets that map text into the semantic frames it evokes are important resources for multiple Natural Language Processing (NLP) tasks (Aharon et al. 2010, Das et al. 2014, Shen and Lapata 2007). Among such resources, FrameNet (Baker et al. 1998) is the richest and most descriptive by far, providing a set of more than 1,200 generic semantic frames, as well as over 200,000 annotated sentences in English. However, this kind of resource is expensive and time-consuming to build, since both the definition of the frames and the annotation of sentences require expertise in the underlying knowledge. Furthermore, it is difficult to decide both the granularity and the domains to consider while defining the frames. Thus, such resources only exist for a reduced amount of languages (Boas and (ed.) 2009) and even English lacks domain-specific resources in multiple domains. Contrasting with frames and their slots, semantic roles are limited and more generic. Furthermore, although their number is not consensual in the literature, there is a set of core semantic roles which is common to every theory Palmer et al. (2005), (2017). However, textual data annotated for semantic roles is also rare for less common languages.

An approach to alleviate the effort in the process of building semantic frame resources is to induce the frames evoked by a collection of documents using unsupervised approaches. However, most research on this subject is focused on verb arguments and the induction of their semantic roles (e.g. Lang and Lapata 2014, Titov and Klementiev 2012, Titov and Khoddam 2015) or on the induction of semantic frames from verbs with two arguments (e.g. Materna 2012, Ustalov et al. 2018). To address this issue and to define a benchmark for future research, a shared task was proposed in the context of SemEval 2019 (QasemiZadeh et al. 2019). This task was focused on the unsupervised induction of FrameNet-like frames through the grouping of verbs and their arguments according to the requirements of three different subtasks. The first one focused on clustering instances of verbs according to the semantic frame they evoke while the others focused on clustering the arguments of those verbs, both according to the frame-specific slots they fill and the generic semantic roles they play.

In a previous study (Ribeiro et al. 2019), we approached the first subtask from a network perspective. More specifically, we applied a graph-clustering approach to a network with contextualized representations of verb instances as nodes, to identify communities of verb instances that evoke the same frame. Furthermore, we controlled the granularity of the frames using a distance threshold for edge creation. That is, we connected two nodes in the network with an edge if the cosine distance between them was below a certain threshold.

In the present work, we extend that study to the remaining subtasks by using a similar approach to identify groups of verb arguments that have the same semantic role and combining them with the semantic frame evoked by the corresponding verb to identify groups of arguments that fill the same frame-specific slot. Furthermore, we explore different approaches to generate the contextualized representations of both verb instances and their arguments, including their combination. Finally, we also compare the performance of multiple community detection algorithms.

In the remainder of the paper, we start by providing an overview of previous approaches to the unsupervised induction of semantic frames and semantic roles, in "Related work". Then, in "Semantic frame induction approach", we describe our semantic frame induction approach. "Experimental setup" describes our experimental setup, including the used dataset, the evaluation approach, and implementation details. The results of our experiments are presented and discussed in "Results & discussion". Finally, in "Conclusions" section, we summarize the contributions of this article and provide pointers for future work.

Related work

Before the shared task in the context of SemEval 2019, there were already some approaches to unsupervised semantic frame induction. For instance, LDA-Frames (Materna 2012) relied on topic modeling and, more specifically, on Latent Dirichlet Allocation (LDA) (Blei et al. 2003), to jointly induce semantic frames and their frame-specific semantic roles. On the other hand, Ustalov et al. (2018) approached the induction of frames through the triclustering of Subject-Verb-Object (SVO) triples using the Watset fuzzy graph-clustering algorithm (Ustalov et al. 2017), which induces word-sense information in the graph before clustering. However, although these approaches are able to induce semantic frames, they can only be applied to verb instances with certain characteristics, such as a fixed number of arguments.

In comparison to the induction of semantic frames, the unsupervised induction of semantic roles has captured more attention and, consequently, research on that task is more extensive. Still, most of the studies on the task focused on the induction of PropBank-style semantic roles (Palmer et al. 2005). These differ from FrameNet frame slots, since they are centered on verbs instead of semantic frames and there is a small number of predefined argument roles. For instance, Titov and Klementiev (2012) represented arguments using a set of syntactic features and then explored the use of two models based on the Chinese Restaurant Process (Ferguson 1973), achieving similar performance. One of the models induces semantic roles for each predicate independently using an iterative clustering approach, starting with one cluster per argument. The other considers a distance-dependent prior shared among predicates to place the arguments in a similarity graph and uses a label propagation approach to induce the semantic roles. This approach was later generalized by Modi et al. (2012) to the induction of the FrameNet frames evoked by verbs and the frame-specific slots filled by their arguments. However, given the high granularity of FrameNet frames, the performance was lower than that observed for the original application to semantic role labeling, especially for the induction of the semantic frames evoked by the verbs.

Lang and Lapata (2014) approached the PropBank-style semantic role labeling task using a graph partitioning approach over a multilayer graph. Each layer corresponds to a feature, that is, each pair of nodes, which correspond to the arguments, is connected through multiple edges, each corresponding to their similarity according to that feature.

Then, two clustering approaches were considered, achieving similar results. The first is an adaptation of agglomerative clustering to the multilayer setting. Instead of combining the similarity values into a single score, it clusters the arguments in each layer and then combines the obtained scores into a multilayer score. Clusters with greater multilayer similarity are then merged together, with larger clusters being prioritized. The second clustering approach consists of propagating cluster membership along the graph edges until convergence.

In contrast to he previous approaches, Titov and Khoddam (2015) proposed a reconstruction-error maximization framework which comprises two main components: an auto-encoder, responsible for labeling arguments with induced roles, and a reconstruction model, which takes the induced roles and predicts the argument that fills each role, that is, it tries to reconstruct the input. The learning error is obtained by comparing the reconstructed argument to the original one. This enables the use of a larger feature set and more complex features, similarly to supervised approaches that typically perform better.

Since we are approaching the subtasks defined in the context of SemEval 2009 Task 2: Unsupervised Lexical Semantic Frame Induction (QasemiZadeh et al. 2019), it is important to provide an overview on the task and to describe the competing approaches in further detail. Overall, the task focused on the unsupervised induction of the FrameNet-like frames evoked by sentences extracted from the Penn Treebank 3.0 PTB (Marcus et al. 1993) corpus. More specifically, it focused on the grouping of the verb instances present in those sentences, as well as of the arguments of those verbs, according to the requirements of three different subtasks. The first one focused on clustering instances of verbs according to the semantic frame they evoke while the others focused on clustering the arguments, both according to the frame-specific slots they fill and the generic semantic roles they play. While the gold standard for semantic frames and their slots was based on a subset of the frames defined in FrameNet (Baker et al. 1998), the generic semantic role annotations used the VerbNet (Palmer et al. 2017) set of labels.

Starting with the subtask of clustering verb instances into semantic frame heads, Arefyev et al. (2019) outperformed the competition using a two-step agglomerative clustering approach. First, it generates a small set of large clusters containing instances of verbs which have at least one sense that evokes the same frame. Then, the verb instances of each cluster are clustered again to distinguish the different frames that are evoked according to the different senses. In both steps, the generation of the representations of the instances relies on BERT (Devlin et al. 2019). Nonetheless, while the first step relies on the contextualized representation given by an empirically selected layer of the model, the second step uses BERT as a language model to generate possible context words that provide cues for the sense of the verb instance. To do so, multiple Hearst-like patterns (Hearst 1992) are applied to the sentence in which the verb instance occurs and the context words correspond to those generated to fill the slots in the patterns. The representation of the instance is then given by a TF-IDF-weighted average of the representations of the most probable context words. The number of clusters in the first step was obtained by performing local optimization on the development data while clustering the development and test data together. In the second step, clusters with less than 20 instances or containing specific undisclosed verbs were left intact. In the remainder, the number of clusters was selected to maximize the silhouette score.

Anwar et al. (2019) used a more simplistic approach based on the agglomerative clustering of contextualized representations of the verb instances. The number of clusters was defined empirically. In the system submitted for participation in the competition, the contextualized representations were obtained by concatenating the context-free representation of the verb instance obtained using Word2Vec (Mikolov et al. 2013) with the TF-IDF-weighted average of the representations of the remaining words in the sentence. However, in a post-evaluation experiment, better results were achieved using the mean of contextualized representations generated by ELMo (Peters et al. 2018).

Finally, Ribeiro et al. (2019) also relied on contextualized representations of the verb instances, but used a graph-based approach. They experimented with both the sum of the representations generated by ELMo (Peters et al. 2018) and those generated by the last layer of the BERT model (Devlin et al. 2019). Better results were achieved with the former. The contextualized representations are used as the nodes in a graph and connected by a distance-weighted edge if the cosine distance between them is below a given threshold based on a function of the mean and standard deviation of the pairwise distances between the nodes. Finally, the graph clustering algorithm Chinese Whispers (Biemann 2006) is applied to the graph to identify communities of nodes that evoke the same frame. This approach achieved high performance on the development data, but did not generalize well to the test data.

For the subtask of clustering verb arguments into generic semantic roles, both Ribeiro et al. (2019) and Anwar et al. (2019) achieved their best results by training a logistic regressor on the development data. As features, both used embedding representations of the arguments – BERT (Devlin et al. 2019) and Word2Vec (Mikolov et al. 2013), respectively – and handcrafted morphosyntactic features. However, since these are supervised approaches, they did not qualify for the task. Anwar et al. (2019) explored the application of agglomerative clustering to the same set of features, but were outperformed by Ribeiro et al. (2019), who used the same approach as for clustering verb instances into semantic frame heads, but with a different function for computing the edge creation threshold. Finally, for the subtask of clustering verb arguments into semantic frame slots, all the participants combined the clusters obtained for the other two tasks, which assumes that frame slots are simply semantic roles in context.

Semantic frame induction approach

Our approach for clustering verb instances into semantic frame heads and verb arguments into the corresponding semantic roles is summarized in Algorithm 1. It builds on and generalizes the approach used by Ribeiro et al. (2019) to compete in the shared task in the context of SemEval, introducing a set of key modifications to improve performance and the ability to generalize. Overall, it starts by generating a contextualized representation of each verb or argument instance. These representations are then used as nodes in a network/graph in which each pair of nodes is connected through an edge if the distance between them is below a certain threshold that controls granularity. Finally, a community detection algorithm is applied to the graph to identify groups of verb instances that evoke the same frame or groups of verb arguments that play the same semantic role.

Below, we describe the steps of the algorithm in further detail. However, before proceeding, it is important to make some remarks regarding the clustering of verb arguments into generic semantic roles. In contrast to frames, which may vary in terms of granularity and domain according to the context in which they are used, VerbNet semantic roles are limited in number and can be seen as generic. Thus, given a sufficient amount of labeled data, approaching the task in a supervised fashion seems more appropriate. This was confirmed in the context of SemEval, since both Anwar et al. (2019) and Arefyev et al. ((Arefyev et al. 2019)) surpassed the unsupervised approaches using logistic regression. However, labeled data is not always available, especially for low-resource languages and domain specific use cases. Thus, approaching the problem in an unsupervised fashion is still relevant. For that reason and for consistency with the shared task in the context of SemEval, we explore the use of the same base approach for clustering verb instances into semantic frame heads and their arguments into generic semantic roles. For grouping arguments into semantic frame slots, we pair the verb and argument clusters. As stated in "Related work" section, this approach assumes that slots are simply semantic roles in context, which is not always true, but is a good approximation.

Contextualized representation

The use of contextualized word representations has led to state-of-the-art performance on multiple NLP tasks (Devlin et al. 2019). These improve traditional uncontextualized word embeddings (e.g. Bojanowski et al. 2017, Mikolov et al. 2013, Pennington et al. 2014) by including information regarding the whole segment in which a word appears in its representation. That is, in contrast to uncontextualized approaches, which generate the same representation for every occurrence of a word, contextualized approaches generate a different representation for occurences of the same word in different contexts. This context information is particularly important for semantic frame induction, since it allows the distinction between different word senses, which evoke different frames. Consequently, as discussed in "Related work" section, all of the approaches used to compete in the shared task in the context of SemEval relied on contextualized word representations. Anwar et al. (2019) used representations generated by ELMo Peters et al. (2018), Arefyev et al. (2019) used representations generated by BERT Devlin et al. (2019), and Ribeiro et al. (2019) experimented with both. There are other approaches to generate contextualized word representations, such as GPT (Radford et al. 2018) and XLNet (Yang et al. 2019). Still, ELMo and BERT are two highly representative approaches.

ELMo (Peters et al. 2018) was one of the first approaches dedicated to the generation of contextualized word representations. It extends the traditional uncontextualized word representation approaches by passing the generated context-free representations through a stack of two bi-directional Long Short-Term (LSTM) units (Hochreiter and Schmidhuber 1997), which capture the dependencies between each word and those that surround it. The word representations generated by ELMo provide information at three levels: the context-free representation of the word and context information at two levels, given by the output of each LSTM. The authors have shown that one of these levels typically provides information concerning the semantic sense of the word, while the other is more related to syntax. Since these two levels modify the context-free representation and the value range of the latter is typically wider than those of the context levels, the context information can be summed to the context-free representation to obtain variations of the word representation according to the context.

Instead of relying on LSTMs, BERT (Devlin et al. 2019) is based on the Transformer architecture (Vaswani et al. 2017) and currently leads to state-of-the-art results on multiple benchmark NLP tasks. It uses token, segment, and positional embeddings as input and a variable number of self-attention layers in both its encoder and decoder. When provided a sequence of tokens, it outputs contextualized representations of each word, as well as a combined representation for the whole sequence. The latter can be connected directly to a classification layer, allowing the weights of the model to be fine-tuned to a specific task. Additionally, the contextualized representations of each word can be obtained from any of the self-attention layers. However, in contrast to ELMo, there is no established relation between the representations generated by each of BERT's layers and specific kinds of information. Thus, a common approach is to use the representations generated by the last self-attention layer of the decoder, since they contain information from all the layers that precede it.

Although the representations generated by BERT are typically seen as state-of-the-art word representations, Ribeiro et al. (2019) observed higher performance in their experiments when using ELMo representations. This is probably due to the fact that BERT representations are not in a linear space in which the cosine distance is appropriate. In fact, Arefyev et al. (2019) noticed that BERT tends to generate representations of the different forms of the same lexeme that are distant in terms of both Euclidean and cosine distances. They tried to identify a distance metric that was appropriate for correlating such representations, but were unsuccessful.

For coverage, we explore the use of contextualized word representations generated by both ELMo and BERT. However, given the higher performance achieved using ELMo in previous studies on the task, we perform more thorough experiments to assess the information that they are able to provide. In addition to the combination of the information provided by the three levels of ELMo representations, we also explore the use of each level independently. This way, we are able to assess which information is actually important for the task and if it varies for the clustering of verb instances into semantic frame heads and its arguments into semantic roles.

To generate the contextualized representation of multi-word verb instances or arguments, we use a dependency parser to generate the dependency tree of the corresponding segment. Then, we identify the head word of the instance, that is, the shallowest of the words that belong to the instance in the tree, and use the corresponding contextualized representation. From a dependency-based perspective, this word can be seen as a representative of the instance and it is the one which has direct relations to other elements in the segment. Furthermore, since we are using representations that capture the context surrounding each word, the representation of the head word also includes information regarding the other words in the instance and is independent from language-specific grammar rules.

Since the generic semantic roles describe the roles played by the arguments with respect to the action or state described by the verb that evokes the semantic frame, information regarding the verb can provide cues for the unsupervised induction of semantic roles. The contextualized representation of the arguments already includes some information regarding the verb. However, the importance of the relation between the arguments and the verb can be made more explicit by also considering the representation of the verb. The compositionality and geometrical properties that are typically observed among word vectors are some of their most promising characteristics (Mikolov et al. 2013). We rely on them in our experiments combining the representations of the arguments with those of the corresponding verb. More specifically, we perform experiments in which we sum or subtract the representation of the verb from that of the argument. The intuition behind these operations is that the sum represents the combination of the verb and argument and the subtraction leaves the dependency between them. Additionally, we also consider a simple concatenation of both representations, which increases the dimensionality of the embedding space.

There are more complex approaches to combine the representations of verbs and their arguments. For instance, Modi and Titov (2014) used a neural model to generate event embeddings from the representations of the verb and arguments that describe the event and used them in a script modeling task (Modi 2016). Since these event embeddings represent the whole situation described in a segment, they are more appropriate for semantic frame induction than semantic role induction. They also require supervised training for a certain task, which is outside the scope of this work. Thus, we leave the exploration of additional compositional approaches for future work, together with the fine-tuning of the ELMo and BERT representations to similarity tasks.

Network creation

In order to approach semantic frame induction as a community detection problem, we must place the verb instances or their arguments in a graph $G = (V, E)$, with V corresponding to the set of nodes and E to the set of edges, weighted according to a function $w : E \to \boldsymbol{R}$. Thus, we start by creating a node for each verb instance or argument to cluster.

To create the edges, we start by calculating the distance between the contextualized representations of each pair of nodes $\left(v, v'\right)$ in G. In Algorithm 1, we kept the distance function generic to show that any distance metric can be used in the approach. However, in our experiments we use the cosine distance, that is, $D_{v,v'} = 1 - \cos\left(\theta_{v,v'}\right) : \left(v, v'\right) \in V^2, v \neq v'$, with $\theta_{v,v'}$ being the angle between v and v'. Among the most common

Algorithm 1 Semantic Frame Induction Approach

```
Input: $S$ // The set of sentences
Input: $I$ // The set of instances to cluster
Input: FINDHEAD // The function that finds the head token of an instance
Input: EMBED // The contextualized representation approach
Input: DISTANCE // The distance function
Input: SIMILARITY // The similarity function
Input: $d$ // The neighboring threshold
Input: DETECTCOMMUNITIES // The community detection algorithm
Output: $C$ // The set of clusters
  $V \leftarrow \{\text{EMBED}(S_i, \text{FINDHEAD}(i)) : i \in I\}$
  $D \leftarrow \left\{(v, v', \text{DISTANCE}(v, v')) : \left(v, v'\right) \in V^2, v \neq v'\right\}$
  $W \leftarrow \left\{(v, v', \text{SIMILARITY}\left(v, v'\right) : \left(v, v'\right) \in V^2, v \neq v'\right\}$
  $E \leftarrow \left\{\left(v, v', w\right) : \left(v, v', w\right) \in W, D_{v,v'} < d\right\}$ // $w$ is the weight of the edge
  $G \leftarrow (V, E)$
  $C \leftarrow \text{DETECTCOMMUNITIES}(G)$
  return $C$
```

distance metrics between word vectors, we opted for the cosine distance in detriment of the Euclidean distance, since the cosine distance is bounded and the magnitude of word vectors is typically related to the number of occurrences. Thus, the angle between the vectors is a better indicator of the semantic differences between the words. Furthermore, the Euclidean distance has issues in spaces with high dimensionality (Aggarwal et al. 2001, Domingos 2012). Still, we performed preliminary experiments to confirm that using the cosine distance leads to better results than the Euclidean distance.

Then, we define the set of edges E by connecting pairs of nodes $\left(v, v'\right)$ if the distance between them is below a certain threshold d, that is, $D_{v,v'} < d$. The definition of this threshold is particularly important, since it controls the granularity of the induced frames. Having control over this granularity is important, since it allows us to induce more specific or more abstract frames, both of which are relevant in different scenarios. Furthermore, this control allows us to define granularity in a small set of instances and then induce frames with a similar granularity in a different set. The latter was the main issue of Ribeiro et al. (2019) approach at the shared task in the context of SemEval, whose performance on the development set did not generalize to the test set. That happened since the threshold was selected using a function of the statistics of the distribution of pairwise distances, which vary according to the contexts covered by the datasets and the number of instances. Hence, and since the test set covered a broader set of contexts, applying the same function on the development and test sets led to the generation of frames with different granularity. We fix this issue by defining the threshold through local optimization on the development set and then using the same fixed threshold across sets.

The distance threshold for edge creation discards the direct connections between nodes that are not close enough when targeting a specific granularity. Still, nodes that are more similar to each other are expected to be more strongly related to each other. Thus, we weight the edges between two nodes using a similarity function, attributing higher weight to edges between more similar nodes. The similarity function is typically inversely correlated to the distance function. However, in Algorithm 1, we kept it generic to show that it does not necessarily need to be the similarity counterpart of the distance metric. Still, we use the cosine similarity in our experiments, that is, given an edge between two nodes $\left(v, v'\right)$, the weight of that edge is given by $W_{v,v'} = \cos\left(\theta_{v,v'}\right)$, with $\theta_{v,v'}$ being the angle between v and v'.

Community detection

Given a network and the need of identifying groups of nodes that share a set of properties, community detection, or graph clustering, is one of the most used methods. The concept is simple: to group sets of nodes that are densely connected between them. As a result, the network is divided in clusters that help us to classify the nodes based on the communities they belong to. Community detection is a very well-known problem but without universal solution (Fortunato and Hric 2016, Schaub et al. 2017). Depending on the properties of the networks and of the algorithms used, the resulting communities can show significant differences. Although there is a considerable amount of community detection/graph clustering algorithms already available, we can only consider those which are able to deal with weighted networks and that do not require a predefined number of communities. Given these two criteria, we explore three different algorithms: Chinese Whispers (Biemann 2006), Louvain Method (Blondel et al. 2008), and Label Propagation (Cordasco and Gargano 2010). However, for assessing the relevance of weighting the edges, we also explore the use of Clauset et al. (2004) Greedy Modularity algorithm, which does not take the weights into account.

We start by exploring the use of Chinese Whispers (Biemann 2006), which has already been used for semantic frame induction Ribeiro et al. (2019, 2019). Furthermore, previous studies have shown that it is able to handle clusters of different sizes, scales well to large graphs, and typically outperforms other clustering approaches on NLP tasks (Biemann 2006, Ustalov et al. 2017). It is a simple but effective graph-clustering algorithm based on the idea that nodes that broadcast the same message to their neighbors should be aggregated. It starts by attributing each node to a different cluster. Then, in each iteration, the nodes are processed in random order and are attributed to the cluster with the highest sum of edge weights in their neighborhood. Thus, more importance is given to edges with higher weight. This process is repeated until there are no changes or the maximum number of iterations is reached.

The Louvain Method (Blondel et al. 2008) (or Louvain Modularity) is a greedy optimization algorithm that runs in two phases, aiming to optimize the modularity of a partition of a weighted network. Modularity is a quantity that measures the fraction of the edges in the network that connect nodes of the same type (i.e., within-community edges) minus the expected value of the same quantity in a network with the same community divisions but random connections between the nodes (Newman 2004). The modularity can be either positive or negative, with positive values indicating the possible presence of community structure (Newman 2006). In the first phase of the Louvain Method, each node is itself a community and the method starts by optimizing the modularity locally, looking for maximal modularity between neighbors, generating small communities. This process ends when it reaches a local maximum, that is, no other combination can improve the modularity. In the second phase it builds a new network in which the nodes are the communities defined in the previous step. These two phases are repeated iteratively until the maximum modularity is achieved and a hierarchy of communities is produced.

Similarly to Chinese Whispers, the Label Propagation algorithm by Cordasco and Gargano (2010) uses the network structure as a guide to detect communities, by propagating labels along the edges. In the beginning, each node is given a label. Then, at each time step, each node performs an update function that consists of adopting the label that the majority of its neighbors has. In a weighted graph, this update function takes into account the weights of the edges, meaning that a higher weight means the label appears more often. During this update, if there is more than one possible choice, the node chooses its next label randomly. This iterative process stops when no node changes its label.

Experimental setup

In this section we describe our experimental setup in terms of dataset and evaluation approach. Furthermore, we provide implementation details that allow future reproduction of our experiments.

Dataset

In our experiments, we use the same dataset used in the context of SemEval 2009 Task 2: Unsupervised Lexical Semantic Frame Induction (QasemiZadeh et al. 2019). This dataset consists of sentences extracted from the PTB (Marcus et al. 1993) with verbs annotated with FrameNet (Baker et al. 1998) frames and arguments annotated with frame slots and generic semantic roles using the VerbNet format (Palmer et al. 2017). The development set consists of 600 verb instances with 1,211 arguments labeled for both semantic role and frame slot. These were extracted from 588 sentences and comprise 41 frames, 20 semantic roles, and 102 frame slots. The test set consists of 4,620 verb instances with 9,466 arguments labeled for semantic role and 9,510 for frame slot. These were extracted from 3,346 sentences and comprise 149 frames, 32 semantic roles, and 436 frame slots.

Additionally, all the sentences in the dataset are annotated with morphosyntactic information in the CoNLL-U format (Buchholz and Marsi 2006).

Evaluation approach

For direct comparison with the approaches that competed in the shared task in the context of SemEval 2019, we evaluate our approach using the same metrics used in that task, namely Purity $\mathrm{F_1}$ (Steinbach et al. 2000) and BCubed $\mathrm{F_1}$ (Bagga and Baldwin 1998). The former is the harmonic mean of purity and inverse-purity:

$$\text{Purity } \mathrm{F_1} = \frac{2 * \text{Purity} * \text{inverse-Purity}}{\text{Purity} + \text{inverse-Purity}} \tag{1}$$

where the purity is given by

$$\text{Purity} = \frac{1}{N} \sum_{c \in C} \max_{g \in G} |c \cap g| \tag{2}$$

where N is the number of instances, C is the set of clusters generated by the system, and G is the set of gold-standard clusters. Conversely, the inverse-purity, also called collocation, is given by

$$\text{inverse-Purity} = \frac{1}{N} \sum_{g \in G} \max_{c \in C} |c \cap g| \tag{3}$$

Thus, purity metrics focus on the quality of each cluster independently. On the other hand, BCubed metrics focus on the distribution of instances of the same category across the clusters. BCubed $\mathrm{F_1}$ is the harmonic mean of BCubed precision and recall:

$$\text{BCubed } \mathrm{F_1} = \frac{2 * \text{BCubed Precision} * \text{BCubed Recall}}{\text{BCubed Precision} + \text{BCubed Recall}} \tag{4}$$

where BCubed precision is given by

$$\text{BCubed Precision} = \frac{1}{N} \sum_{c \in C} \frac{1}{|c|} \sum_{g \in G} |c \cap g|^2 \tag{5}$$

and BCubed recall is given by

$$\text{BCubed Recall} = \frac{1}{N} \sum_{g \in G} \frac{1}{|g|} \sum_{c \in C} |c \cap g|^2 \tag{6}$$

Additionally, we report the number of induced clusters. Since some of the community detection algorithms we use are nondeterministic, the values we report for these metrics refer to the mean and standard deviation over 30 runs.

Since we are approaching the problem from a network-based perspective, we also report the number of edges and the clustering coefficient of the network corresponding to the neighboring threshold with highest performance in each scenario.

In addition to the approaches that competed in the shared task in the context of SemEval, we also compare our approach with a set of baselines that consists of generating one cluster per verb lemma for the semantic frame head induction task, one cluster per argument-verb dependency label for semantic role induction, and the pairing of these two for frame slot induction.

Implementation details

Starting with the contextualized representation of verb instances and their arguments, to obtain ELMo representations, we used the original model (Peters et al. 2018), as provided by the AllenNLP package (Gardner et al. 2017), which was trained as a bi-directional language model on the 1 Billion Word Benchmark (Chelba et al. 2014). For each instance, we generated the contextualized embeddings for the corresponding sentence and then selected the representations of the head token of the instance. The representation is then given by three vectors of dimensionality 1,024, corresponding to the context-free representation of the head token and the two levels of context information. To obtain BERT representations, we used the large uncased model provided by its authors (Devlin et al. 2019), which was trained on both the BooksCorpus (Zhu et al. 2015) and the English Wikipedia, not only as a masked language model, also referred to as a Cloze task (Taylor 1953), but also for a next sentence prediction task. Since, as discussed in the "Contextualized representation" section, there is no established relation between the representations generated by each of BERT's layers and specific kinds of information, we use the representations generated by the last self-attention layer of the decoder, which contain information from all the layers that precede it. In the model we use, this layer also produces a vector of dimensionality 1,024.

Regarding the community detection algorithms, to apply Chinese Whispers, we relied on (Ustalov and et al. 2018) implementation. We did not use weight regularization and performed a maximum of 20 iterations. To apply the Louvain Method, we relied on (Aynaud 2009) implementation, with randomization activated. To apply the Greedy Modularity and Label Propagation algorithms, we used the implementation provided by the NetworkX package (Hagberg et al. 2004), with the default parameters.

Finally, to obtain the syntactic dependencies used to determine the head token of multiword verb instances and arguments, we used the annotations provided with the dataset, which were obtained automatically using a dependency parser.

Results & discussion

Before starting the presentation and discussion of the results, it is important to make some remarks: first, in order to limit the number of experiments, we performed them incrementally. For instance, we only experimented with different community detection algorithms after identifying the approach for generating contextualized representations that leads to the highest performance when using Chinese Whispers. Consequently, the presentation of the results for each subtask is structured to follow this incremental approach.

Additionally, we structure this section by starting with the results achieved when clustering verb instances into semantic frame heads, followed by those achieved when clustering arguments into semantic roles, and finishing with the clustering of arguments into semantic frame slots, since it is the combination of the other two. However, we combine the results of the three subtasks in a last section, for comparison with the results reported in previous studies.

Finally, regarding the actual presentation of the results, since the contextualized representations may have negative components, the cosine distance varies in the interval [0, 2]. However, to improve readability and since using higher neighboring thresholds does not lead to changes in the results, we limit the plots shown in this section to the interval [0, 1]. Regarding the tables, unless stated otherwise, the results they report are those achieved using the neighboring threshold that leads to the highest performance in terms of BCubed F_1.

Clustering verb instances into semantic frame heads

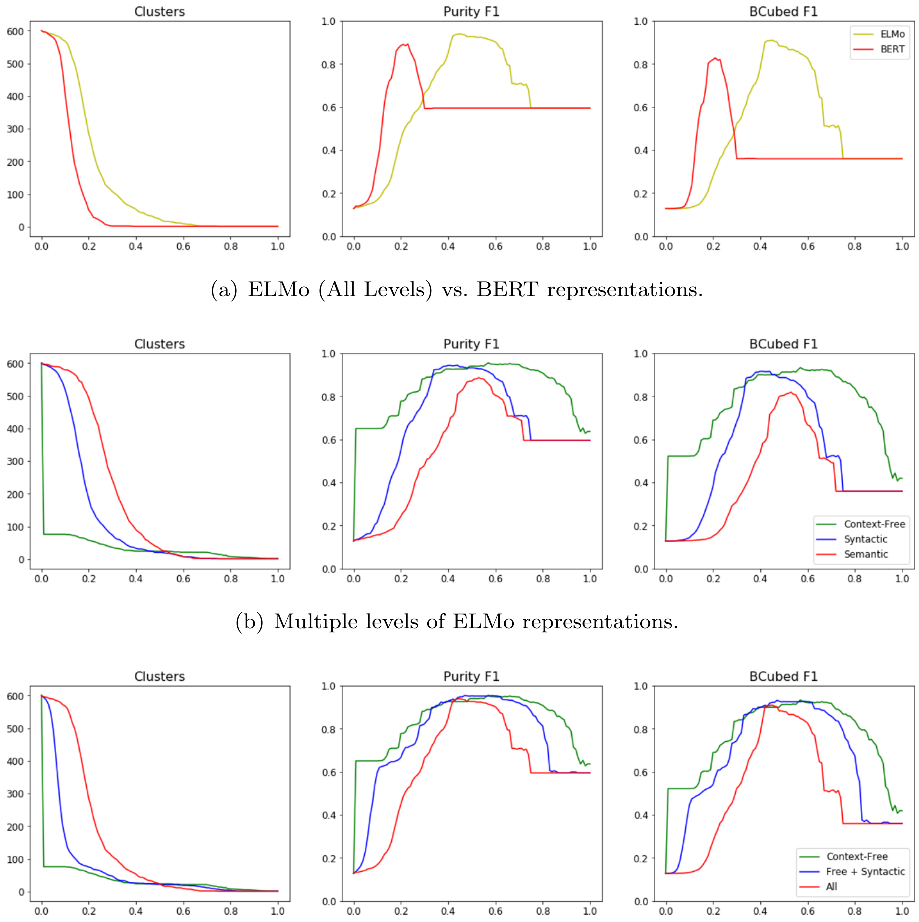
For readability, we structure the presentation of the results on this subtask according to the incremental experiments that we performed on the development data. We start by discussing the impact of using different representations of the verb, then we discuss the weighting of the edges, and, finally, we compare the performance of different community detection algorithms. Last, we discuss the ability of the approach to generalize to the test data and perform cluster analysis.

Verb representation

Starting with the contextualized representation of verb instances, both Fig. 1a and Table 1 show that, as expected, ELMo representations lead to higher performance than BERT representations, since the latter are not in a linear space in which the cosine distance is appropriate. However, when considering the neighboring thresholds that lead to the best results, the number of clusters generated when using BERT representations agrees with the number of frames, while it is underestimated when using ELMo representations. On the one hand, this confirms that the cosine similarity between BERT representations fails to capture verb instances with similar semantics. On the other hand, it suggests that when using ELMo representations, the difficulties arise when attempting to perform more fine-grained distinctions. Finally, ELMo representations are more robust to changes in the neighboring threshold, as revealed by a wider interval with reduced decrease in performance around the threshold with highest performance.

Regarding the information provided by the multiple levels included in ELMo representations, in Fig. 1b and the second block of Table 1, we can see that, independently, the context-free representation is the most informative of the three and the most robust to changes in the threshold. The initial drop in the number of clusters is due to its lack of context information, which makes all the instances of the same verb become connected as soon as the threshold is higher than zero.

The lower performance of the levels that provide context information on their own was expected, since they represent changes in the word sense of the verb according to the context, but lack information regarding the verb itself, which is important for the identification of semantic frames. Comparing the performance of both levels, we can see that the level that typically captures the semantic context leads to worse performance than that which captures syntactic context and even harms performance in combination with the other levels. However, this can be explained by the fact that the ELMo model was trained as a bi-directional language model. Thus, it focuses on generating representations that allow the identification of the most probable words that follow or precede a given



(a) ELMo (All Levels) vs. BERT representations.

(b) Multiple levels of ELMo representations.

(c) Combinations of multiple levels of ELMo representations.

Fig. 1 Semantic frame induction results achieved using different contextualized representations of verb instances. The horizontal axes refer to the neighboring threshold used to create the edges

sequence, which is not directly related to the evoked semantic frames. Furthermore, the semantic context layer is the closest to the output layer and, consequently, it is more prone to overfitting to this task. On the other hand, the syntactic context is more generic and, since the sense of a verb can be related to the syntactic tree in which it occurs, it provides information that is relevant for the task.

Table 1 Semantic frame induction results achieved using different contextualized representations of verb instances

	d	Edges	CC	Clusters	Purity F_1	BCubed F_1
BERT	0.21	18,945	0.66	41.13±0.56	89.03±1.02	82.76±1.22
Context-Free	0.57	39,441	0.97	22.03±0.31	95.57±0.58	93.35±0.71
Syntactic Context	0.41	27,201	0.80	30.93±0.25	94.32±0.16	91.65±0.20
Semantic Context	0.53	20,660	0.67	24.47±0.52	88.64±0.33	81.92±0.54
Free + Syntactic	0.47	37,913	0.94	22.97±0.41	**95.83±0.28**	**93.66±0.32**
All	0.45	21,448	0.73	34.73±0.51	93.93±0.30	91.04±0.59

The first row refers to BERT representations, the second block to the different levels of ELMo representations, and the last block to combinations of those levels. *d* refers to the neighboring threshold. *CC* refers to the clustering coefficient

Table 2 Semantic frame induction results according to the weighting of the edges

	d	Edges	CC	Clusters	Purity F_1	BCubed F_1
Weighted	0.47	37,913	0.94	22.97±0.41	**95.83±0.28**	**93.66±0.32**
Unweighted	0.46	37,415	0.93	22.90±0.30	95.77±0.16	93.56±0.31

d refers to the neighboring threshold. *CC* refers to the clustering coefficient

As shown in Fig. 1c and the third block of Table 1, the highest performance is achieved when using the combination of the context-free representation and the syntactic context. Still, the average increase in BCubed F_1 in relation to when using the context-free representation on its own is of just 0.33 percentage points, which suggests that the context information is only able to disambiguate a reduced amount of specific cases. However, the threshold that leads to the highest performance in the combination is lower. This means that the graph has less edges and consequently, is less connected. Still, the number of clusters, around 23, is nearly half of the number of frames in the gold standard, 41, which means that the graph should be even less connected. As previously discussed, since the performance decreases for lower thresholds, this suggests that problems occur when performing more fine-grained distinctions. Thus, either the representations or the distance metric are unable to capture all the information required to group the instances in FrameNet-like frames.

Edge weighting

Regarding the weighting of the edges, the results in Table 2 show that the difference in average top performance is of just 0.06 and 0.10 percentage points in terms of Purity F_1 and BCubed F_1, respectively, which is not significant. This shows that the presence or absence of the edges is more important for the approach than their weight. In fact, if the neighboring threshold for creating the edges was not considered, then all the nodes would be merged into a single cluster, regardless of whether the edges were weighted or not. Still, in Fig. 2, we can see that using weighted edges slightly increases the robustness of the approach to changes in the neighboring threshold. Consequently, we kept them in subsequent experiments.

Community detection

Regarding community detection algorithms, the results in Table 3 show that the top performance of the Greedy Modularity algorithm, which is the only one which does not consider the weights, is 8.68 and 13.21 percentage points below that of the remaining algorithms in terms Purity F_1 and BCubed F_1, respectively. However, considering the

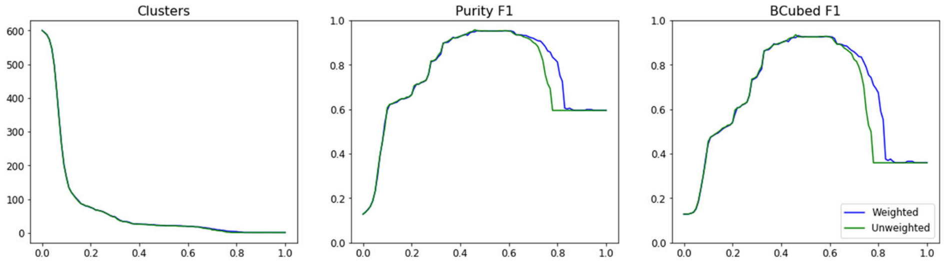


Fig. 2 Semantic frame induction results according to the weighting of the edges. The horizontal axes refer to the neighboring threshold used to create the edges

Table 3 Semantic frame induction results achieved using different community detection algorithms

	d	Edges	CC	Clusters	Purity F_1	BCubed F_1
Chinese Whispers	0.47	37,913	0.94	22.97±0.41	**95.83±0.28**	**93.66±0.32**
Louvain Method	0.43	35,491	0.92	23.00±0.00	**95.83±0.00**	**93.66±0.00**
Label Propagation	0.41	33,711	0.90	23.00±0.00	**95.83±0.00**	**93.66±0.00**
Greedy Modularity	0.31	21,753	0.84	33.00±0.00	87.15±0.00	80.45±0.00

d refers to the neighboring threshold. *CC* refers to the clustering coefficient

results reported in "Edge weighting" section regarding our experiments with weighted and unweighted edges, we assume that the lower performance is not due to the fact that the algorithm does not consider the weights, but rather because it is not appropriate for the task. This inappropriateness is further revealed in Fig. 3, since the Greedy Modularity algorithm leads to irregular patterns as the neighboring threshold increases, even in terms of the number of clusters. On the other hand, the remaining algorithms lead to similar patterns, except for the higher end of the neighboring threshold, in which the Louvain Method seems more robust. Still, on average, the top performance of the three algorithms is the same, with some runs of the Chinese Whispers algorithm leading to higher performance.

Results on test data

Figure 4 shows the results achieved by applying the top performing approaches to the test data. Although the performance is lower, we can observe patterns similar to those observed on the development data. The only difference is that there is a slightly more pronounced performance drop immediately after the threshold that leads to highest performance. Nonetheless, as shown in Table 4, the thresholds selected on development data are lower and close to the best threshold on test data. This shows that using local optimization to define the neighboring threshold leads to an appropriate generalization of the granularity of the frames. The largest difference, 0.07, is observed when using the Label Propagation algorithm, which is also the one with the largest performance drop, 0.93 and 1.19 percentage points in terms of Purity F_1 and BCubed F_1, respectively, when comparing the use of the development threshold and the best threshold for the test set. On the other hand, the Chinese Whispers algorithm is the one that generalizes better, achieving the highest performance on the test set, even when considering the runs with lowest performance, and a difference of just 0.29 percentage points in terms of average Purity F_1 and 0.36 percentage points in terms of BCubed F_1, when comparing the use of the development threshold and the best threshold for the test set.

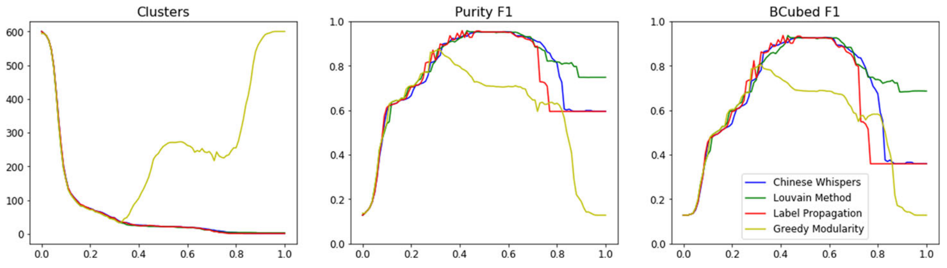


Fig. 3 Semantic frame induction results using different community detection algorithms. The horizontal axes refer to the neighboring threshold used to create the edges

Table 4 Semantic frame induction results on the test data

	d	Edges	CC	Clusters	Purity F_1	BCubed F_1
CW Dev. Threshold	0.47	347,202	0.91	196.63±1.68	79.97±0.21	73.07±0.25
CW Best Threshold	0.49	364,829	0.91	186.33±0.98	**80.26±0.17**	**73.43±0.19**
LM Dev. Threshold	0.43	308,781	0.89	198.00±0.00	79.38±0.00	72.40±0.00
LM Best Threshold	0.46	338,312	0.90	181.00±0.00	79.68±0.00	72.89±0.00
LP Dev. Threshold	0.41	288,351	0.88	227.00±0.00	78.74±0.00	71.48±0.00
LP Best Threshold	0.48	355,911	0.91	186.00±0.00	79.67±0.00	72.67±0.00

d refers to the neighboring threshold. *CC* refers to the clustering coefficient. *CW*, *LM*, and *LP* refer to Chinese Whispers, Louvain Method, and Label Propagation, respectively

Contrasting with what happened in the development data, the approach overestimates the number of clusters. However, this can be explained by the fact that the test data includes more instances of different verbs that evoke the same frame. Once again, this suggests that either the representations of the verb instances or the distance metric are unable to capture all the required information. To assess this, we performed some additional error analysis.

Figure 5 shows the evolution of BCubed precision and recall, as well as purity and inverse purity, as the neighboring threshold increases. We can see that, as expected, since the number of clusters decreases with the neighboring threshold, the precision also decreases, while the recall increases. More interesting is the fact that, before the threshold of highest performance, precision decreases slowly while recall increases fast and, after that threshold, precision starts decreasing fast. This suggests that many clusters of verb instances that evoke different semantic frames are merged after that threshold, which supports the claim that the difficulties arise when attempting to perform more fine-grained distinctions.

By inspecting the generated clusters, we have identified a set of common errors. First of all, while according to the annotations there are at least 2 verb instances that evoke each of the 149 frames, among the generated clusters there are 30 which contain a single instance. All of these correspond to outliers of larger clusters, which explains the overestimation of the number of clusters.

Another common error is the merging of verb instances that evoke semantic frames that are only distinguishable by the type of the arguments. For instance, the semantic frames *building* and *manufacturing* can both be evoked by instances of the verb *to build*. However, the first is evoked when the object argument of the verb is a building and the second when it is, for instance, a type of vehicle. Among others, a similar situation occurs for the

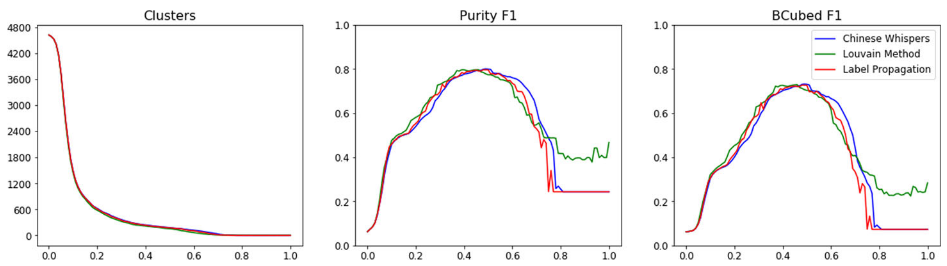


Fig. 4 Semantic frame induction results on the test data. The horizontal axes refer to the neighboring threshold used to create the edges

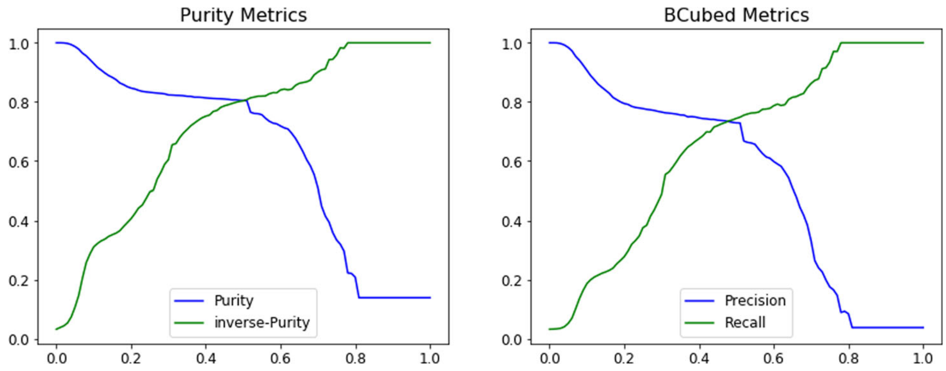


Fig. 5 Precision (purity) and recall (inverse purity) of the top performing semantic frame induction approach on the test data. The horizontal axes refer to the neighboring threshold used to create the edges

activity start and *process start* semantic frames. The inability to distinguish between verb instances that evoke these frames suggests that the contextualized representations of the verb instances are not capturing enough information regarding the arguments. This is an issue that can be approached by fine-tuning the representations for the task or by generating combined embeddings for the verb and its arguments, such as those proposed by Modi and Titov (2014).

The last common error that we identified is the inability to distinguish between verb instances that evoke a semantic frame and others that refer to the cause of those semantic frames. This problem occurs, for instance, between the *activity start* and *cause to start* semantic frames, as well as between *change position on a scale* and *cause change of position on a scale*. Many of these cases are hard to distinguish, even for humans. Thus, the only possible solution that we are able to propose is to check whether fine-tuning the representations to sentence similarity tasks can lead to improved performance in these situations.

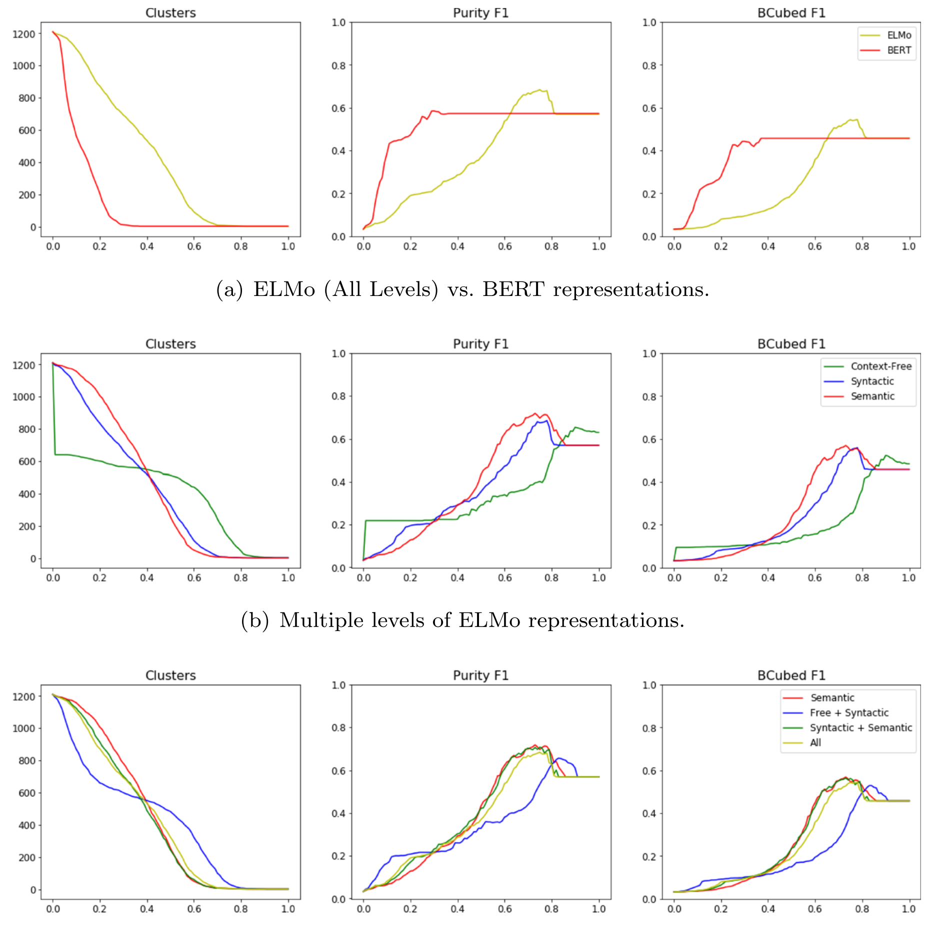
There are other less common types of error, such as verb instances that evoke the same semantic frame distribution across multiple clusters, and others which are situational and for which it is difficult to identify a generic cause.

Clustering arguments into semantic roles

Similarly to the previous subtask, we structure the presentation of the results on this subtask according to the incremental experiments that we performed on the development data. In terms of structure, the only difference in relation to the previous is that after discussing the representation of the arguments, we discuss the experiments in which we also included information regarding the verb.

Argument representation

Starting with the contextualized representation of the arguments, in Fig. 6a and Table 5, we can see that, similarly to what happened when clustering verb instances into semantic frame heads, using ELMo representations leads to higher performance than using BERT representations. However, in this case, both lead to a severe underestimation of the number of clusters, which, consequently, impairs the performance in relation to that observed when clustering verb instances.



(a) ELMo (All Levels) vs. BERT representations.

(b) Multiple levels of ELMo representations.

(c) Combinations of multiple levels of ELMo representations.

Fig. 6 Semantic role induction results achieved using different contextualized representations of verb arguments. The horizontal axes refer to the neighboring threshold used to create the edges

Regarding the information provided by the multiple levels included in ELMo representations, in Fig. 6b and the second block of Table 5, we can see that, in this case, the context-free representation is the less informative of the three. This was expected, considering that the semantic roles are generic and, thus, they are typically not related to specific words, but rather to the dependencies between the arguments and the verb. Furthermore,

Table 5 Semantic role induction results achieved using different contextualized representations of verb arguments

	d	Edges	CC	Clusters	Purity F_1	BCubed F_1
BERT	0.37	348,761	0.81	2.00±0.00	57.27±0.00	45.69±0.00
Context-Free	0.90	157,154	0.55	4.47±0.88	65.30±0.69	52.28±0.76
Syntactic Context	0.78	146,978	0.57	3.57±0.88	68.44±5.34	55.85±4.13
Semantic Context	0.73	97,442	0.56	5.93±1.00	**71.88±1.61**	**56.91±1.94**
Free + Syntactic	0.84	140,768	0.56	3.80±0.98	65.41±1.05	52.89±0.77
Syntactic + Semantic	0.73	101,180	0.57	5.63±0.80	70.66±3.29	56.36±2.11
All	0.78	143,273	0.57	4.40±0.61	67.97±3.50	54.41±2.82

The first row refers to BERT representations, the second block to the different levels of ELMo representations, and the third block to combinations of those levels. *d* refers to the neighboring threshold. *CC* refers to the clustering coefficient

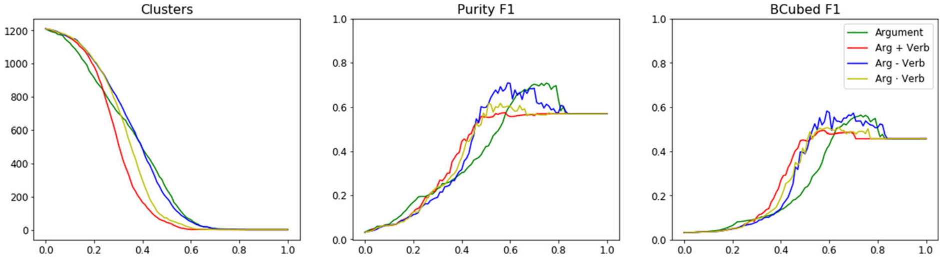


Fig. 7 Semantic role induction results achieved using information regarding the verb. The horizontal axes refer to the neighboring threshold used to create the edges

in this case, independently, the level that captures semantic context leads to the highest performance and more robustness to changes in the neighboring threshold in comparison to the remaining levels. This makes sense considering that the verbs and their arguments are typically sequential in the segments that contain them. Thus, the representation of the arguments generated by the semantic context layer contain information regarding their relation to the verb, which is highly related to the semantic roles they play.

Figure 6c and the third block of Table 5 show that including the context-free information is always harmful, even in combination with the remaining levels. On the other hand, the combination of the levels that provide syntactic and semantic context leads to similar performance to that achieved using the semantic context level on its own. Since the combination considers additional information, we used it in subsequent experiments, in an attempt to improve the ability of the approach to generalize to scenarios in which the semantic roles are played in different contexts.

Verb information

As discussed in "Contextualized representation" section, the generic semantic roles describe the roles played by the arguments with respect to the action or state described by the verb that evokes the semantic frame. Thus, information regarding the verb can provide cues for the unsupervised induction of semantic roles. Figure 7 and Table 6 show the results achieved in the experiments in which we explicitly included information regarding the verb. We can see that both the sum and the concatenation of the verb instance representation to that of the argument still lead to an underestimation of the number of clusters and reduce the performance in relation to when using the representation of the argument on its own. On the other hand, the subtraction of the verb representation from that of the argument leads to an overestimation of the number of clusters and a performance improvement of 1.84 percentage points in terms of BCubed F_1. This confirms that this subtraction operation is actually able to isolate the relation between the argument and the verb. Furthermore, the results suggest that this representation is more robust to the non-determinism of the Chinese Whispers algorithm. Overall, even though

Table 6 Semantic role induction results achieved using information regarding the verb

	d	**Edges**	**CC**	**Clusters**	**Purity F_1**	**BCubed F_1**
Sum	0.58	77,388	0.57	5.00±0.00	57.33±0.09	49.40±0.10
Subtraction	0.59	30,635	0.55	56.33±0.94	**71.88±0.14**	**58.20±0.06**
Concatenation	0.64	93,029	0.59	3.33±0.47	60.64±1.43	50.82±1.52

d refers to the neighboring threshold. *CC* refers to the clustering coefficient

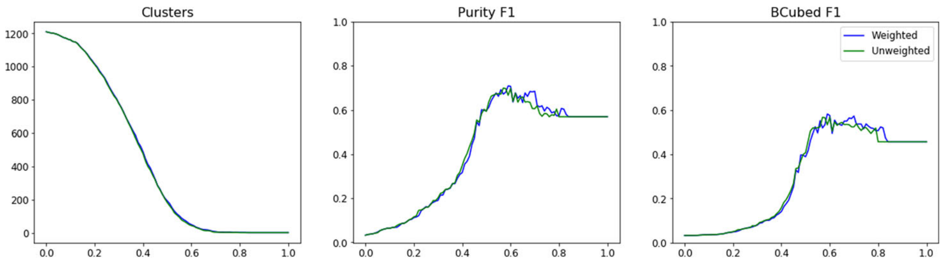


Fig. 8 Semantic role induction results according to the weighting of the edges. The horizontal axes refer to the neighboring threshold used to create the edges

this representation is more discriminative than that of the argument on its own, the top performance on the development set is 58.20% in terms of BCubed F$_1$, which shows that there is still a significant amount of semantic information that it is not able to capture or that the distance metric is not the most appropriate for capturing semantic similarity.

Edge weighting

Figure 8 shows that, similarly to what happens when clustering verb instances, if the weights of the edges are not considered, the patterns observed as the neighboring threshold increases are similar to when using a weighted graph. However, Table 7 shows that, in this case, the difference in top performance is significant, with a decrease of 1.11 and 1.48 percentage points in terms of average Purity F$_1$ and BCubed F$_1$, respectively, when the weights are not considered. Also similarly to the task of clustering verb instances according to the semantic frame they evoke, the presence or absence of the edges is more important for the approach than their weight and all the arguments are merged in a single cluster if the neighboring threshold is not considered. However, in this case, the weighting of the edges provides some additional information that is relevant for the unsupervised induction of semantic roles.

Community detection

Regarding the community detection algorithms, in Fig. 9 and Table 8, we can see that, once again, the Greedy Modularity algorithm is that with lower performance and the one which leads to the most distinct evolution patterns as the neighboring threshold increases, especially regarding the number of clusters. As for the remaining algorithms, in Fig. 9 we can see that, from a high-level perspective, all of them follow similar patterns as the neighboring threshold increases. However, the Louvain Method has a smoother drop after the threshold of highest performance. Furthermore, Table 8 shows that, in contrast to what happened when clustering verb instances, the Chinese Whispers algorithm outperforms the remaining by at least 5.52 percentage points in terms of Purity F$_1$ and 4.45 in terms of BCubed F$_1$. Also, the top performance of the Louvain Method and Label Propagation approaches is achieved when the number of clusters is underestimated. If we

Table 7 Semantic role induction results according to the weighting of the edges

	d	Edges	CC	Clusters	Purity F$_1$	BCubed F$_1$
Weighted	0.59	30,635	0.55	56.33±0.94	**70.88±0.14**	**58.20±0.06**
Unweighted	0.60	34,164	0.56	43.33±0.94	69.77±0.65	56.72±1.10

d refers to the neighboring threshold. *CC* refers to the clustering coefficient

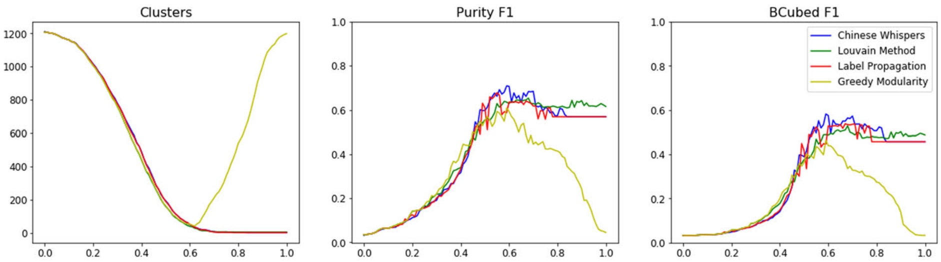


Fig. 9 Semantic role induction results achieved using different community detection algorithms. The horizontal axis refers to the neighboring threshold used to create the edges

take a closer look into the evolution in terms of Purity F_1 and BCubed F_1 as the neighboring threshold increases, we can see that it is actually noisy, with several oscillations around the thresholds of highest performance. This noisy evolution may explain the differences in the top performance of the approaches which had equal performance on the verb instance clustering task.

Results on test data

Although Chinese Whispers outperformed the remaining community detection algorithms on the development data, for consistency with the experiments regarding the clustering of verb instances according to the semantic frame they evoke, we also assessed the performance of the Louvain Method and the Label Propagation algorithm on the test data. In Fig. 10, we can see that while Chinese Whispers and the Label Propagation algorithm follow patterns similar to those observed on the development set, the Louvain Method follows a distinct pattern, exacerbating the slight difference observed on the development set. Furthermore, as shown in Table 9, it is the one with the largest difference, 0.17, between the top performing neighboring thresholds on the development and test data. However, the highest difference in performance when comparing the use of both thresholds is observed for the Label Propagation algorithm, with a difference of 8.90 and 7.54 percentage points in terms of Purity F_1 and BCubed F_1, respectively. On the other hand, when using Chinese Whispers, the corresponding differences are of just 0.51 and 0.65 percentage points, which, once again, reveals its ability to generalize. However, when using the development threshold, the number of clusters is further overestimated.

Analyzing the best approach in more detail, in Fig. 11, we can see that, *contrasting with* what happened in the verb instance clustering task, precision and recall decrease and increase at similar velocities, respectively. Nonetheless, in this case, there is a noisy evolution with several oscillations after the threshold of highest performance. Still, since the development threshold is lower than the best threshold for the test data, that noisy evolution has no impact on the task.

Table 8 Semantic role induction results achieved using different community detection algorithms

	d	Edges	CC	Clusters	Purity F_1	BCubed F_1
Chinese Whispers	0.59	30,635	0.55	56.33±0.94	**70.88±0.14**	**58.20±0.06**
Louvain Method	0.68	77,659	0.59	9.33±0.47	65.36±0.08	52.84±0.24
Label Propagation	0.66	63,742	0.58	15.00±0.00	64.00±0.00	53.75±0.00
Greedy Modularity	0.59	30,635	0.55	54.00±0.00	60.94±0.00	46.45±0.00

d refers to the neighboring threshold. *CC* refers to the clustering coefficient

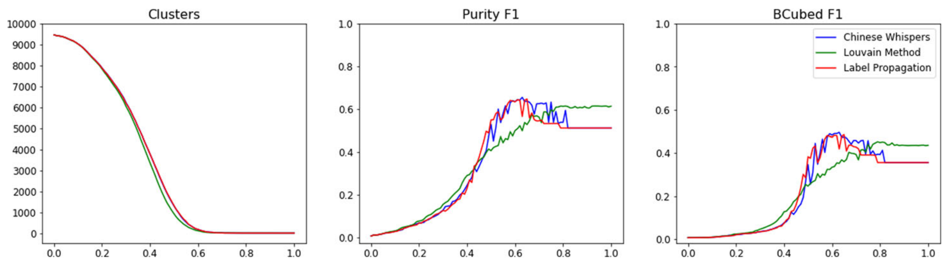


Fig. 10 Semantic role induction results on the test data. The horizontal axes refer to the neighboring threshold used to create the edges

By inspecting the generated clusters we noticed that, similarly to what happened in the verb instance clustering task, the overestimation of the number of clusters is partially explained by single-instance clusters containing outliers of larger clusters.

Additionally, the approach clearly fails to distinguish arguments that play the *co-theme* and *topic* roles from those that play the more generic *theme* role. The *co-theme* role is attributed to arguments when there are multiple *themes* in a semantic frame and all of them participate equally. Thus, a possible solution to this problem is to also consider information regarding the remaining arguments. However, it must be distinguishable from that regarding the argument that is being focused. A *topic* is a type of *theme* that is specific to verbs of communication. Thus, from the community detection perspective, it actually corresponds to a sub-community. The community detection algorithms we explored are not able to identify sub-communities directly. A possible approach to this problem is to perform a second community detection step, in which the algorithm is applied to the members of each of the communities detected in the first step.

The remaining problems with the generated clusters refer to the distribution of arguments that play the same role among several clusters, as well as the merging of arguments that play several different roles in the same cluster. This confirms that, as the results on the development data suggested, either the representations of the arguments or the distance metric are not able to capture all the similarity information required to induce generic semantic roles. Thus, we intend to check whether fine-tuning the representations to sentence similarity tasks can lead to improved performance on semantic role induction.

Clustering arguments into semantic frame slots

Since our approach for inducing semantic frame slots consists of combining the induced semantic role for an argument with the semantic frame evoked by the corresponding verb,

Table 9 Semantic role induction results on the test data

	d	Edges	CC	Clusters	Purity F_1	BCubed F_1
CW Dev. Threshold	0.59	525,265	0.47	207.33±1.25	64.96±0.26	48.85±0.33
CW Best Threshold	0.63	884,614	0.47	75.00±2.45	**65.47±0.58**	**49.50±0.66**
LM Dev. Threshold	0.68	1,725,277	0.47	13.33±0.47	55.12±1.81	38.64±1.40
LM Best Threshold	0.85	14,195,334	0.60	4.00±0.00	60.57±0.45	43.72±0.70
LP Dev. Threshold	0.66	1,319,897	0.47	38.00±0.00	55.83±0.00	40.92±0.00
LP Best Threshold	0.65	1,154,382	0.47	43.00±0.00	64.73±0.00	48.46±0.00

d refers to the neighboring threshold. *CC* refers to the clustering coefficient. *CW*, *LM*, and *LP* refer to Chinese Whispers, Louvain Method, and Label Propagation, respectively

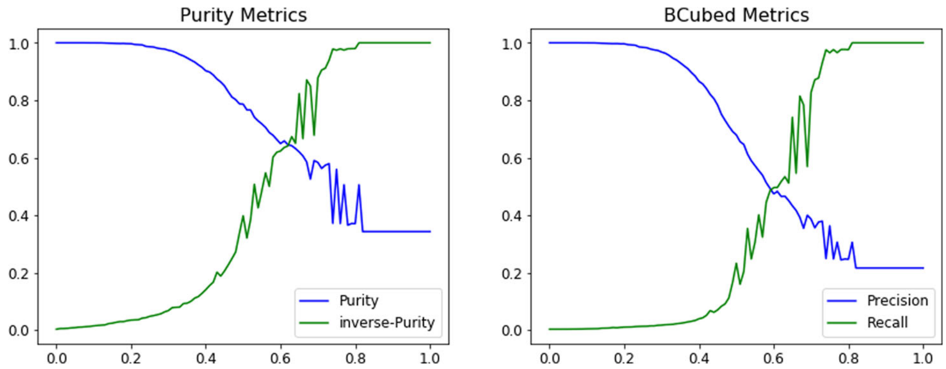


Fig. 11 Precision (purity) and recall (inverse purity) of the top performing semantic role induction approach on the test data. The horizontal axes refer to the neighboring threshold used to create the edges

this discussion only covers the results achieved using the top performing approaches to each of those tasks.

In the first row of Table 10, we can see that the number of clusters induced on the development set is close to the gold standard of 102. However, even though the semantic frame induction approach has high performance on this set, the overall performance is impaired by the lower performance on semantic role induction.

The remaining rows of Table 10 show the results achieved on the test set, using either the neighboring thresholds computed on the development data, or the thresholds that led to highest performance on the test sets for semantic frame and semantic role induction. It is interesting to observe that the highest performance is achieved when using the development thresholds, which supports the claim that our approach generalizes well.

Comparison with previous approaches

Although there were already some approaches to unsupervised semantic frame induction before the shared task in the context of SemEval 2019 (QasemiZadeh et al. 2019), we cannot compare them to ours directly, since they can only be applied to verb instances with certain characteristics, such as a fixed number of arguments. Similar restrictions occur for previous approaches on semantic role induction. Thus, we only compare our results with those of the approaches that competed in that shared task.

Starting with the clustering of verb instances into the semantic frames they evoke, the approach with the highest performance in the competition was that by Arefyev et al. (2019). As described in "Related work" section, it is a two-step clustering approach of contextualized representations generated by BERT, which starts by generating a small set of clusters containing instances of verbs which have at least one sense that evokes the same frame and then clusters the verb instances of each of those clusters independently to distinguish the different frames that are evoked according to the different senses. Anwar

Table 10 Semantic frame slot induction results

	Clusters	Purity F_1	BCubed F_1
Development	96.33±4.89	80.57±2.34	73.91±2.58
Test (Dev. Thresholds)	852.20±61.99	66.70±2.26	56.76±2.38
Test (Test Thresholds)	613.30±65.33	65.95±2.18	55.39±1.69

Table 11 Comparison of semantic frame induction results with those of previous approaches in terms of performance on the test data

	Purity F_1	**BCubed F_1**
Baseline	73.78	65.35
Ribeiro et al. (2019)	75.25	65.32
Anwar et al. (2019)	76.68	68.10
Arefyev et al. (2019)	78.15	70.70
Our Approach (Dev. Threshold)	**79.97**	**73.07**

et al. (2019) used a more simplistic approach based on the agglomerative clustering of contextualized representations of the verb instances, with the number of clusters defined empirically.

Table 11 shows the results of those approaches in comparison to ours and the one-frame-per-verb-lemma baseline. First of all, it is important to note that while Ribeiro et al. (2019) approach, on which ours is based, performed similarly to the baseline in terms of BCubed F_1, the updated approach described in this article outperforms it by 4.37 and 7.72 percentage points in terms of Purity F_1 and BCubed F_1, respectively. This shows the importance of discarding the semantic context provided in the ELMo representations and, most importantly, of identifying a neighboring threshold that allows the approach to generalize. Furthermore, our approach also outperforms the more complex approach by Arefyev et al. (2019) by 2.37 percentage points in terms of BCubed F_1. Consequently, it achieves the current state-of-the-art performance on the task.

Moving to the subtask of clustering verb arguments into semantic roles, both Arefyev et al. (2019) and Anwar et al. (2019) achieved their best results by training a logistic regressor on the development data. As features, both used embedding representations of the arguments, as well as handcrafted morphosyntactic features. However, since these are supervised approaches, they did not qualify for the task. Thus, Anwar et al. (2019) also explored the application of agglomerative clustering to the same set of features.

Table 12 shows the results of those approaches in comparison to ours and the one-role-per-dependency-label baseline. We can see that our approach outperforms the baseline by 8.91 percentage points in terms of Purity F_1 and 9.82 percentage points in terms of BCubed F_1. Furthermore, it outperforms (Ribeiro et al. 2019), which was the winner of the competition, by 3.19 percentage points in terms of BCubed F_1, achieving the current state-of-the-art performance on unsupervised semantic role induction on this dataset. However, the performance is still under 50% and 15.19 percentage points below that of Arefyev et al. (2019) supervised approach. This confirms that the identification of the semantic roles of verb arguments is easier to approach as a

Table 12 Comparison of semantic role induction results with those of previous approaches in terms of performance on the test data

	Purity F_1	**BCubed F_1**
Baseline	56.05	39.03
Ribeiro et al. (2019)	64.10	45.66
Anwar et al. (2019)	62.00	42.10
Arefyev et al. (2019)	~~77.47~~	~~64.04~~
Our Approach (Dev. Threshold)	**64.96**	**48.85**

The results of (Arefyev et al. 2019) are crossed because they were obtained using a supervised approach supervised problem and that better representations and similarity metrics are required for their unsupervised induction.

Finally, for the subtask of clustering verb arguments into semantic frame slots, all the participants of the SemEval shared task combined the clusters obtained for the other two tasks, which assumes that frame slots are simply semantic roles in context. Table 13 compares the performance of our approach with that of those systems, as well as the combination of the baselines for the other two tasks. We can see, once again, that our approach outperforms the winner of the competition. In this case, the improvement is 4.60 percentage points in terms of Purity F_1 and 7.47 percentage points in terms of BCubed F_1. However, it is still outperformed by (Arefyev et al. 2019) approach, which is supervised for the induction of semantic roles. Since the difference in performance is lower than that observed for semantic role induction, we expect our semantic frame slot induction approach to perform significantly better if the semantic roles of the arguments are more accurately predicted.

Conclusions

In this article we have approached the unsupervised induction of semantic frames and semantic roles as community detection problems applied to networks with the verb instances or arguments as nodes, with two nodes connected by an edge if the cosine distance between their contextualized representation is below a threshold that defines the granularity of the induced frames or semantic roles. Conversely, the similarity between contextualized representations is used to weight the edges, with a higher weight attributed to edges between more similar nodes.

We have shown that when clustering verb instances into semantic frame heads, the best performance is achieved when using contextualized representations given by the combination of the context-free and syntactic context levels of ELMo representations (Peters et al. 2018). Complementing the context-free representation with context information allows the distinction of polysemic verbs, which evoke different frames according to the context (Rumshisky and Batiukova 2008). On the other hand, when clustering verb arguments into semantic roles, a higher performance is achieved by discarding the context-free representation and relying solely on the contextualized representations given by the combination of the syntactic and semantic context levels. The context-free representation has a negative impact in this scenario since semantic roles are not related to specific words, but rather to the dependencies between the verbs and the arguments. Consequently, the highest performance on this task was achieved by focusing even more on those dependencies, through the subtraction of the verb instance representation from that of the argument.

Table 13 Comparison of semantic frame slot induction results with those of previous approaches in terms of performance on the test data

	Purity F_1	**BCubed F_1**
Baseline	57.99	45.79
Ribeiro et al. (2019)	50.99	42.75
Anwar et al. (2019)	62.10	49.49
Arefyev et al. (2019)	~~73.11~~	~~64.43~~
Our Approach (Dev. Thresholds)	**66.70**	**56.76**

The results of Arefyev et al. (Arefyev et al. 2019) are crossed because they were obtained using a supervised approach

Additionally, we have observed that the weighting of the edges is not as important as their existence, but it can make the approach more robust to changes in the neighboring threshold. Furthermore, among the community detection algorithms explored in our study, Chinese Whispers (Biemann 2006) leads to the highest performance and generalization ability. This is consistent with previous studies which revealed the high performance of Chinese Whispers on unsupervised NLP tasks (Biemann 2006, Ustalov et al. 2017).

We have performed our experiments on the benchmark dataset defined in the context of SemEval 2019 Task 2 (QasemiZadeh et al. 2019), which allows us to compare our results with those of previous approaches. In this context, the most important step is to identify the threshold that defines correct granularity according to the gold standard annotations. We did so by performing local optimization on the development data and used the same fixed threshold on the test data. This way, we solved the main issue of the approach on which ours was based, which was its lack of generalization ability. In fact, the difference between the best threshold on the development set and that which would lead to the best performance on the test set was of just 0.02 when clustering verb instances into semantic frame heads and 0.04 when clustering arguments into semantic roles. Furthermore, when clustering arguments into semantic frame slots, which we did by combining the semantic role of the argument with the semantic frame evoked by the corresponding verb, the highest performance was achieved when using the thresholds computed on the development data.

Using this approach, we were able to outperform the winners of each subtask in the context of the SemEval shared task. More specifically, it outperformed Arefyev et al. (2019) more complex approach to cluster verb instances into semantic frame heads by 2.37 percentage points in terms of BCubed F_1. Furthermore, it outperformed Ribeiro et al. (2019) approach to semantic role induction and Anwar et al. (2019) approach to semantic frame slot induction by 3.19 and 7.47 percentage points, respectively. Thus, our approach achieves the current state-of-the-art performance on unsupervised semantic frame induction.

Although we were able to outperform all the previous approaches on the task, the 73.07% BCubed F_1 score achieved on semantic frame induction on the test data shows that the approach is not able to capture all the information required to induce FrameNet-like frames and that there is still room for improvement. The 48.85% BCubed F_1 score achieved on semantic role induction in comparison to the 64.04% achieved by Arefyev et al. (2019) using a supervised approach reveals the difficulty of approaching this task in an unsupervised fashion and the need for better contextualized representations or similarity metrics. In this context, the most straightforward approach that can be explored in the future is the fine-tuning of the ELMo and BERT (Devlin et al. 2019) models to sentence similarity tasks, in an attempt to generate contextualized representations that are more appropriate for semantic frame and semantic role induction. In fact, Reimers and Gurevych (2019) have shown that tuning the BERT model to such tasks leads to the generation of sentence representations that can be compared in terms of cosine similarity. Thus, it is possible that the same will occur for the contextualized word representations that it generates.

Another possible path that can be explored in the future is the use of multilayer or multiplex networks Kivelä et al. (2014), either using the same features as (Lang and Lapata

2014), the multiple levels of ELMo representations, or the combination of several syntactic and semantic conceptual associations between words that have been used to build multilayer networks in the context of other computational linguistics tasks. For instance, Massimo et al. Stella et al. (2018) proposed a multiplex network representation of a mental lexicon of word similarities to investigate large-scale cognitive patterns, where each layer represents the semantics, phonology, and taxonomy of the English lexicon, and identified a cluster of words which are used with greater frequency, are identified, memorized, and learned more easily, and have more meanings than expected at random. This cluster is the largest viable cluster across all layers. In another application, (Siew and Vitevitch 2019) studied the interplay between orthographic influence on spoken word recognition and phonological influence on visual word recognition, creating a phonographic network language, in which links are placed between words if they are both phonologically and orthographically similar to each other, that is, if they overlap in both the phonological and orthographic layers. The advantage of using networks of this kind is that there are distance metrics which have been shown to be informative in terms of semantic similarity (Kenett et al. 2017).

By analyzing the clusters that were generated in the context of the semantic frame induction task, we noticed that the approach is currently unable to distinguish verb instances that evoke different semantic frames that are only distinguishable by the type of the arguments. This suggests that the contextualized representations of the verb instances are not capturing enough information regarding the arguments. Thus, a possible approach to address this issue is generating combined embeddings for the verb and its arguments, such as the event embeddings proposed by Modi and Titov (2014). However, similarly to fine-tuning the ELMo and BERT models, these embeddings must be trained for a subsequent task in a supervised fashion.

On the other hand, by analyzing the clusters that were generated in the context of the semantic role induction task, we noticed that one of the problems of the approach is its inability to distinguish arguments that play a semantic role that is a specialization of another, more generic, role. A possible approach to this problem that can be explored in the future is to perform a second community detection step, in which the algorithm is applied to the members of each of the communities detected in the first step, in order to identify sub-communities. This has similarities to the two-step approach proposed by Arefyev et al. (2019) for semantic frame induction.

Still regarding semantic roles, since there is a set of core semantic roles that are common across most theories, we also want to explore their recognition in a supervised fashion, similarly to Arefyev et al. (2019). This way, we can also improve the performance of our approach to the induction of semantic frame slots. Furthermore, we can explore the use of our graph-based approach in a semi-supervised fashion. That is, we create a network with both the development and test instances, initialize the labels of the development nodes with the corresponding semantic roles, and then let them propagate across the whole network.

Finally, another direction that can be explored in the future concerns the incrementality of the approach and its ability to identify the semantic frames evoked by new verb instances and the semantic roles played by their arguments. In this context, the process of updating the network in an incremental fashion is similar to creating the whole network at once. That is, given the contextualized representation of a new instance, the corresponding node can be added to the network. Then, the distance between the contextualized representation of that instance and those of all the other nodes in the network must be computed in order to identify its neighbors, according to the granularity threshold d. Finally, the corresponding weighted edges between the new node and its neighbors are added to the network.

As the network grows in size, computing the distance between a new node and all the others in the network becomes a more expensive process. Thus, it may be necessary to explore processes to reduce the number of distance calculations. A possible approach is to index the nodes as a grid in the representation space, sized according to the granularity threshold. This way, nodes that are not in the same cell or in a neighbor one can be discarded without distance computation. However, for high thresholds, the reduction in number of calculations may not be significant. Another approach is to define a similarity threshold, s, that limits the addition of new nodes to the network. More specifically, if the distance between a new node and any other in the network is below s, then it is considered the same node and is not added to the network. Finally, an additional approach is to compress the network from time to time, by selecting a set of representative nodes for each cluster and discarding the remaining, mapping the edges of the discarded nodes to the representatives. However, the application of the last two approaches, and especially the last one, implies loss of information and may require an updated weighting function, which includes information regarding the number of nodes compressed in a single one. Thus, the applicability of these approaches and the balance between them is a complex problem on its own.

After identifying the communities in a network, identifying the community to which a new node belongs is a simple process. When using an algorithm based on label propagation, the community of a new node can be identified by propagating the labels of its neighbors, similarly to how classes are predicted in weighted nearest neighbors approaches. On the other hand, when using algorithms based on modularity, the community can be identified by calculating the modularity when the new node is attributed to each of the communities its neighbors belong to and selecting that which leads to the highest modularity. However, both cases imply that the communities are fixed after the initial application of the community detection algorithm. That is, there is no contribution from new nodes. In an incremental scenario, the communities are expected to change dynamically. A simple approach to handle this issue is to run the community detection algorithm every time the network changes. Nonetheless, that is a computationally expensive process and, since most community detection algorithms involve some kind of non-determinism, there is the problem of matching the identified communities with those that existed before. Thus, instead of starting from scratch, the approach can start with the previously identified communities, attribute a new label to every new node in the network, and then continue the application of the community detection algorithm until convergence. However, this typically leads to the absorption of the new nodes into the existing communities and, consequently, the identification of new communities rarely occurs. Thus, a balance has to be found between starting from scratch and only propagating the existing communities to the new nodes. This can be done by identifying a set of nodes in the network that are expected to be impacted by the new node and relying solely on that set to update the communities. Approaches for identifying such a set of nodes have been explored in the context of both label propagation (e.g.

Asadi and Ghaderi 2018) and modularity-based (e.g. Zarayeneh and Kalyanaraman 2019) algorithms.

Abbreviations

FCT: Fundação para a Ciência e a Tecnologia; LDA: Latent Dirichlet Allocation; LDC: Linguistic Data Consortium; LSTM: Long Short-Term Memory; NLP: Natural Language Processing; PTB: Penn Treebank 3.0; SVO: Subject-Verb-Object

Authors' contributions

All authors have contributed with ideas for the approach described in the article. ER has performed the experiments. All authors have validated and criticized the results. ER, AST, and RR have structured the article. ER and AST have written the article.

Funding

Eugénio Ribeiro is supported by a PhD scholarship granted by Fundação para a Ciência e a Technologica FCT, with reference SFRH/BD/148142/2019. Additionally, this work was supported by Portuguese national funds through FCT, with references UIDB/50021/2020 and PTDC/EEI-SII/1937/2014, and by PT2020, project number 39703 (AppRecommender).

Availability of data and materials

The dataset that supports the findings of this study is available from the organizers of SemEval 2009 Task 2: Unsupervised Lexical Semantic Frame Induction (QasemiZadeh et al. 2019), but restrictions apply to its availability, since it contains sentences from the PTB (Marcus et al. 1993), which is licenced by the Linguistic Data Consortium LDC.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]INESC-ID, Lisboa, Portugal. [2]Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal. [3]Center for Social and Biomedical Complexity, School of Informatics, Computing, & Engineering, Indiana University, Bloomington, Indiana, USA. [4]Indiana University Network Science Institute (IUNI), Indiana University, Bloomington, Indiana, USA. [5]Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal.

Received: 9 March 2020 Accepted: 26 August 2020
Published online: 22 September 2020

References

Aggarwal CC, Hinneburg A, Keim DA (2001) On the surprising behavior of distance metrics in high dimensional space. In: ICDT. pp 420–434. http://dx.doi.org/10.1007/3-540-44503-X_27

Aharon RB, Szpektor I, Dagan I (2010) Generating entailment rules from framenet. In: ACL, vol. 2. pp 241–246. https://www.aclweb.org/anthology/P10-2045/

Anwar S, Ustalov D, Arefyev N, Ponzetto SP, Biemann C, Panchenko A (2019) HHMM at semeval-2019 task 2: unsupervised frame induction using contextualized word embeddings. In: SemEval. pp 125–129. http://dx.doi.org/10.18653/v1/S19-2018

Arefyev N, Sheludko B, Davletov A, Kharchev D, Nevidomsky A, Panchenko A (2019) Neural granny at semeval-2019 task 2: a combined approach for better modeling of semantic relationships in semantic frame induction. In: SemEval. pp 31–38. http://dx.doi.org/10.18653/v1/S19-2004

Asadi M, Ghaderi F (2018) Incremental community detection in social networks using label propagation method. In: FRUCT. pp 39–47. http://dx.doi.org/10.23919/FRUCT.2018.8588023

Aynaud T (2009) Louvain community detection. GitHub. https://github.com/taynaud/python-louvain/. Accessed 26 Aug 2020

Bagga A, Baldwin B (1998) Algorithms for scoring coreference chains. In: Linguistic Coreference Workshop in the context of the first LREC, Granada. pp 563–566. http://www.lrec-conf.org/lrec1998/

Baker CF, Fillmore CJ, Lowe JB (1998) The berkeley framenet project. In: ACL/COLING, vol. 1. pp 86–90. http://dx.doi.org/10.3115/980451.980860

Biemann C (2006) Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In: Workshop on graph-based methods for natural language processing. pp 73–80. http://dx.doi.org/10.3115/1654758.1654774

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022

Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):10008

Boas HC, (ed.) (2009) Multilingual framenets in computational lexicography: methods and applications. trends in linguistics. studies and monographs, vol. 200. Mouton de Gruyter, Berlin, Germany

Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146

Buchholz S, Marsi E (2006) CoNLL-X shared task on multilingual dependency parsing. In: CoNLL. pp 149–164. http://dx.doi.org/10.3115/1596276.1596305

Chelba C, Mikolov T, Schuster M, Ge Q, Brants T, Koehn P, Robinson T (2014) One billion word benchmark for measuring progress in statistical language modeling. In: INTERSPEECH. pp 2635–2639. https://www.isca-speech.org/archive/interspeech_2014/i14_2635.html

Clauset A, Newman MEJ, Moore C (2004) Finding community structure in very large networks. Phys Rev E 70(6):066111

Cordasco G, Gargano L (2010) Community detection via semi-synchronous label propagation algorithms. In: BASNA. pp 1–8. http://dx.doi.org/10.1109/BASNA.2010.5730298

Das D, Chen D, Martins AFT, Schneider N, Smith NA (2014) Frame-semantic parsing. Computat Linguist 40(1):9–56

Devlin J, Chang M-W, Kenton L, Toutanova K (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT, vol. 1. pp 4171–4186. http://dx.doi.org/10.18653/v1/N19-1423

Domingos P (2012) A few useful things to know about machine learning. Commun ACM 55(10):78–87

Ferguson TS (1973) A bayesian analysis of some nonparametric problems. Ann Stat 1(2):209–230

Fillmore CJ (1976) Frame Semantics and the Nature of Language. Ann N Y Acad Sci 280:20–32

Fortunato S, Hric D (2016) Community detection in networks: a user guide. Phys Rep 659:1–44

Gardner M, Grus J, Neumann M, Tafjord O, Dasigi P, Liu NF, Peters M, Schmitz M, Zettlemoyer LS (2017) AllenNLP: a deep semantic natural language processing platform. CoRR abs/1803.07640. http://dx.doi.org/10.18653/v1/W18-2501

Hagberg A, Schult D, Swart P (2004) NetworkX. GitHub. https://networkx.github.io/. Accessed 26 Aug 2020

Hearst MA (1992) Automatic acquisition of hyponyms from large text corpora. In: COLING, vol. 2. pp 539–545. http://dx.doi.org/10.3115/992133.992154

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Kenett YN, Levi E, Anaki D, Faust M (2017) The semantic distance task: quantifying semantic distance with semantic network path length. J Exp Psychol Learn Mem Cognit 43(9):1470

Kivelä M, Arenas A, Barthelemy M, Gleeson JP, Moreno Y, Porter MA (2014) Multilayer networks. J Compl Netw 2(3):203–271

Lang J, Lapata M (2014) Similarity-driven semantic role induction via graph partitioning. Comput Linguist 40(3):633–670

Marcus M, Santorini B, Marcinkiewicz M (1993) Building a large annotated corpus of english: the penn treebank. Comput Linguis 19(2):330–331

Materna J (2012) LDA-frames: an unsupervised approach to generating semantic frames. In: CICLing. pp 376–387. http://dx.doi.org/10.1007/978-3-642-28604-9_31

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: NIPS. pp 3111–3119. https://dl.acm.org/doi/10.5555/2999792.2999959

Minsky M (1974) A framework for representing knowledge. Technical report, Massachusetts Institute of Technology. https://dspace.mit.edu/handle/1721.1/6089

Modi A (2016) Event embeddings for semantic script modeling. In: CoNLL. pp 75–83. http://dx.doi.org/10.18653/v1/K16-1008

Modi A, Titov I (2014) Inducing neural models of script knowledge. In: CoNLL. pp 49–57. http://dx.doi.org/10.3115/v1/W14-1606

Modi A, Titov I, Klementiev A (2012) Unsupervised induction of frame-semantic representations. In: NAACl-hlt workshop on the induction of linguistic structure. pp 1–7. https://www.aclweb.org/anthology/W12-1901/

Newman MEJ (2004) Analysis of weighted networks. Phys Rev E 70(5):056131

Newman MEJ (2006) Modularity and community structure in networks. Proc Natl Acad Sci 103(23):8577–8582

Palmer M, Bonial C, Hwang JD (2017) Verbnet: capturing english verb behavior, meaning and usage. Oxf Handb Cogn Sci:315–336. http://dx.doi.org/10.1093/oxfordhb/9780199842193.013.15

Palmer M, Gildea D, Kingsbury P (2005) The proposition bank: an annotated corpus of semantic roles. Computat Linguist 31(1):71–106

Pennington J, Socher R, Manning CD (2014) GloVe: global vectors for word representation. In: EMNLP. pp 1532–1543. http://dx.doi.org/10.3115/v1/D14-1162

Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: NAACL-HLT, vol. 1. pp 2227–2237. http://dx.doi.org/10.18653/v1/N18-1202

QasemiZadeh B, Petruck MRL, Stodden R, Kallmeyer L, Candito M (2019) SemEval-2019 task 2: unsupervised lexical frame induction. In: SemEval. pp 16–30. http://dx.doi.org/10.18653/v1/S19-2003

Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training. Preprint. http://openai-assets.s3.amazonaws.com/research-covers/language-unsupervised/language_understanding_paper.pdf.Accessed 26 Aug 2020

Reimers N, Gurevych I (2019) Sentence-bert: sentence embeddings using siamese bert-networks. In: EMNLP-IJCNLP. pp 3973–3983. http://dx.doi.org/10.18653/v1/D19-1410

Ribeiro E, Mendonça V, Ribeiro R, Martins de Matos D, Sardinha A, Santos AL, Coheur L (2019) L2F/Inesc-id at semeval-2019 task 2: unsupervised lexical semantic frame induction using contextualized word representations. In: SemEval. pp 130–136. http://dx.doi.org/10.18653/v1/S19-2019

Ribeiro E, Teixeira AS, Ribeiro R, Martins de Matos D (2019) Semantic frame induction as a community detection problem. In: COMplex networks. pp 274–285. http://dx.doi.org/10.1007/978-3-030-36687-2_23

Rumshisky A, Batiukova O (2008) Polysemy in verbs: systematic relations between senses and their effect on annotation. In: COLINg 2008 workshop on human judgements in computational linguistics. pp 33–41. http://dx.doi.org/10.3115/1611628.1611634

Schaub MT, Delvenne J-C, Rosvall M, Lambiotte R (2017) The many facets of community detection in complex networks. Appl Netw Sci 2(1):1–13

Shen D, Lapata M (2007) Using semantic roles to improve question answering. In: EMNLP-CoNLL. pp 12–21. https://www.aclweb.org/anthology/D07-1002/

Siew CSQ, Vitevitch MS (2019) The phonographic language network: using network science to investigate the phonological and orthographic similarity structure of language. J Exp Psychol Gen 148(3):475

Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. In: KDD Workshop on Text Mining. https://www.cs.cmu.edu/~dunja/PapersWshKDD2000.html

Stella M, Beckage NM, Brede M, De Domenico M (2018) Multiplex model of mental lexicon reveals explosive learning in humans. Sci Rep 8(1):1–11

Taylor WL (1953) Cloze procedure: a new tool for measuring readability. Journal Bull 30(4):415–433

Titov I, Khoddam E (2015) Unsupervised induction of semantic roles within a reconstruction-error minimization framework. In: NAACL-HLT, vol. 1. pp 1–10. http://dx.doi.org/10.3115/v1/N15-1001

Titov I, Klementiev A (2012) A bayesian approach to unsupervised semantic role induction. In: EACL, vol. 1. pp 12–22. https://www.aclweb.org/anthology/E12-1003/

Ustalov D, et al. (2018) Chinese Whispers for Python. GitHub. https://github.com/nlpub/chinese-whispers-python/. Accessed 26 Aug 2020

Ustalov D, Panchenko A, Biemann C (2017) Watset: automatic induction of synsets from a graph of synonyms. In: ACL, vol 1. pp 1579–1590. http://dx.doi.org/10.18653/v1/P17-1145

Ustalov D, Panchenko A, Kutuzov A, Biemann C, Ponzetto SP (2018) Unsupervised semantic frame induction using triclustering. In: ACL, vol. 2. pp 55–62. http://dx.doi.org/10.18653/v1/P18-2010

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: NIPS. pp 5998–6008. https://papers.nips.cc/paper/7181-attention-is-all-you-need

Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV (2019) XLNet: generalized autoregressive pretraining for language understanding. In: NIPS. pp 5753–5763. https://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding

Zarayeneh N, Kalyanaraman A (2019) A fast and efficient incremental approach toward dynamic community detection. In: ASONAM. pp 9–16. http://dx.doi.org/10.1145/3341161.3342877

Zhu Y, Kiros R, Zemel R, Salakhutdinov R, Urtasun R, Torralba A, Fidler S (2015) Aligning books and movies: towards story-like visual explanations by watching movies and reading books. In: ICCV. pp 19–27. http://dx.doi.org/10.1109/ICCV.2015.11

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.