Transductive Visual Verb Sense Disambiguation

Sebastiano Vascon
University of Venice
sebastiano.vascon@unive.it

Sinem Aslan
University of Venice
Ege University
sinem.aslan@unive.it

Gianluca Bigaglia
University of Venice
gianluca.bigaglia@unive.it

Lorenzo Giudice
University of Venice
lorenzo.giudice@unive.it

Marcello Pelillo
University of Venice
marcello.pelillo@unive.it

Abstract

Verb Sense Disambiguation is a well-known task in NLP, the aim is to find the correct sense of a verb in a sentence. Recently, this problem has been extended in a multimodal scenario, by exploiting both textual and visual features of ambiguous verbs leading to a new problem, the Visual Verb Sense Disambiguation (VVSD). Here, the sense of a verb is assigned considering the content of an image paired with it rather than a sentence in which the verb appears. Annotating a dataset for this task is more complex than textual disambiguation, because assigning the correct sense to a pair of <image, verb> requires both non-trivial linguistic and visual skills. In this work, differently from the literature, the VVSD task will be performed in a transductive semi-supervised learning (SSL) setting, in which only a small amount of labeled information is required, reducing tremendously the need for annotated data. The disambiguation process is based on a graph-based label propagation method which takes into account mono or multimodal representations for <image, verb> pairs. Experiments have been carried out on the recently published dataset VerSe, the only available dataset for this task. The achieved results outperform the current state-of-the-art by a large margin while using only a small fraction of labeled samples per sense[1].

1. Introduction

Every language has ambiguous words, e.g., in English, the word *apple* can be referred to as either an IT company, a fruit, or a city. Word Sense Disambiguation (WSD) is a common task in natural language processing [24], where the goal is to automatically recognize the correct sense of a word within a sentence.

Verb Sense Disambiguation (VSD) is a sub-problem of WSD where the correct sense of a *verb* in a sentence is aimed to be identified [35]. For instance, while the most common sense of the verb *run* is the one related to moving quickly, it might have a different sense regarding to its context, such as the one related to machine operations (*the washing machine is running*) or covering a distance (*this train runs hundreds of miles every day*); all these senses share the same verb, but they have quite different meanings.

VSD is an utmost important task, affecting different domains. For example, in an NLP retrieval scenario, it is required the search engine to group the results by senses, hence disambiguate the verb senses in queries to retrieve the correct results [8]. VSD also takes an important role in other NLP tasks such as, machine translation [35], semantic role labeling [1] and question answering [26].

In addition to the typical NLP tasks, VSD can be brought to a Computer Vision (CV) domain, taking into account problems like Action Recognition (AR) and Human Object Interaction (HOI) [32, 7]; the authors exploit the identification of objects and entities in an image to infer either the action that is being performed or the correct verb that links those entities and objects. Even if there are some clear overlappings between VSD and AR/HOI, the latter do not take into account the ambiguity of verbs.

The analogy between these tasks in NLP and CV fields can be exploited by combining the features of both domains to improve a disambiguation system's overall performances. Motivated by this fact, recently, [12] introduced the *Visual Verb Sense Disambiguation* (VVSD). In a VVSD task, the goal is to disambiguate the sense of a verb paired with an image. Differently from a standard NLP disambiguation task, in which the context is provided by a phrase, here the context is provided by an image.

In [12] the authors proposed the first (and only) well curated dataset to assess algorithms' performances for VVSD

[1] Code available: `https://github.com/GiBg1aN/TVVSD`

tasks. The reported baselines comprise both supervised and unsupervised models using both unimodal (textual or visual) and multimodal features (textual and visual). Annotating a dataset for this task is very expensive, since it requires both non-trivial language and visual knowledge [27]. Toward this direction, in this work, we tackle the multimodal VVSD problem, offering a new perspective based on semi-supervised learning which brings significant performance gain at a lower labeling-cost. The strength of SSL algorithms arises when the available labeled set size is not significant to train a fully-supervised classifier or when annotating a full dataset is too expensive or unfeasible. In SSL, only a small amount of labeled data is needed because both labeled and the unlabeled samples embeddings are exploited during inference. Thus, we assume to have a small amount of labeled data (<image,verb> and its sense) to infer the senses of the unlabeled ones.

Among the possible SSL algorithms [41], we choose a game-theoretic model called *Graph Transduction Games* (GTG) [11]. The GTG has been succesfully applied in many different SSL contexts, like deep metric learning [10], matrix factorization [37], image recognition [2], protein-function prediction [39] and, indeed, traditional text-based WSD setting [36]. Moreover, it works consistently better [38] than other graph-based SSL methods like Label Propagation [41] and Gaussian Fields [42].

Our contributions are thus three-fold:

1. We proposed a new model for multimodal visual verb sense disambiguation based on semisupervised learning.
2. We reported an extensive ablation study on the effect of using an increasing number of labeled data.
3. We outperformed the state-of-the-art by a large margin exploiting only a small fraction of labeled data.

2. Related works

Common approaches for WSD/VSD can be grouped into three categories: *supervised*, *unsupervised* and *knowledge-based* methods. Supervised methods [21] rely on sense-tagged corpora which act as the training set. Such algorithms usually exploit linguistic features like: n-grams that surround a word, syntactical dependencies tags (e.g. subject, object, verb) or context information summarized in structures like co-occurrence matrices. Performance of supervised methods is hindered by the requirement of handling all the possible senses of target corpus while it is implausible to have a training set with a sufficiently big sample size for each word/verb sense [24]. Thus, unsupervised learning algorithms, that do not exploit any training data, may be a more suitable solution when the number of senses to handle becomes unfeasible. Purely unsupervised methods rely on the distribution of senses and exploit the fact that words/verbs with the same sense would have similar contextual information. However, while they extract clusters of senses, they do not rely on exact sense labeling of words/verbs which would yield to extracted senses are likely to not match the ones categorized and defined in standard dictionaries. On the other hand at the knowledge-based methods, rather than extracting the sense inventory from the corpus, it is known a-priori. So, a mapping between a dictionary and the occurrences in the corpus is performed and by relying on lexical databases [29, 23, 25] semantic accordance is used to disambiguate. In [36], a semi-supervised learning model for WSD is proposed, facing a pure textual, and not multimodal, task.

While there is a huge literature on Word Sense Disambigutation (WSD) adopting (unimodal) textual features, visual clues for WSD in a multimodal setting was studied by limited works. One of the first approaches is in [3] which used a statistical model based on joint probabilities between images and words. Following an unsupervised approach, [19] applied spectral clustering for image sense disambiguation; while [6] applied co-clustering through textual and visual domain to explore multiple senses of a given noun phrase. [31] used LDA to discover a latent space by exploiting dictionaries definitions to learn distributions that represent senses. A similar task was accomplished in [4] that tried to solve linguistic ambiguities using multimodal data. In [5] they used multimodal data for semantic frame labeling. Performances of all these aforementioned works are quite good, however such techniques are noun-oriented and perform poorly for verb disambiguation tasks. The first attempt to perform a fully verb-oriented sense disambiguation was introduced in [13], which designed a variation of Lesk algorithm [17] that uses the multimodal sense encoding and the multimodal input encoding respectively as the definition and context for the algorithm.

3. Transductive VVSD

In this section we dissect the different components of our model, named *Transductive Visual Verb Sense Disambiguation* (TVVSD). The global picture can be seen in Fig 1.
Our model is made of four steps:

1. Feature extraction for each pair <image, verb>.
2. Construction of a graph-based representation of all pair <image, verb>.
3. Initialization of the assignment between <image, verb> and possible senses.
4. Transductive inference via dynamical system assigning <image, verb> to a sense.

in the following we will consider the i-th pair <image, verb> as an atomic entity.

3.1. Feature extraction

We follow the schema proposed in [12] for a fair comparison with the state-of-the-art, although more recent feature models can be easily plugged in our pipeline. For each pair <image, verb> we extract the following embeddings:
Visual features. Input images are fed into a pre-trained

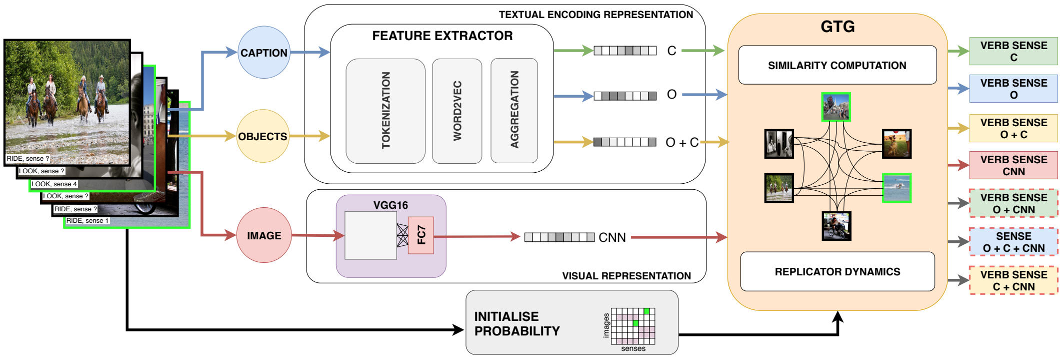


Figure 1: Pipeline of the algorithm considering both labeled (green border) and unlabeled images (black border). The letters O and C stands for features generated from Objects label and Caption.

VGG16[2] model [34], and the output of the last FC layer is used as feature representation, resulting in a vector of 4096 elements for each image. Such vector is then unit-normalized.

Textual features. As in [12], experiments on text have been run on two possible setups: using VerSe textual data annotations (GOLD) or by generating them through state-of-art object detectors and image descriptors (PRED). In the latter scenario, object labels have been predicted using a VGG16 model. Since the VGG16 net classifies images without performing object detection, in [12] they thresholded the output of the SoftMax layer taking only classes that had a score greater than 0.2 (or the highest in case of empty result). This allows to obtain multiple classes/objects per image.

Captions have been generated with NeuralTalk2[3] [40]. For what concerns the encoding, either captions, objects labels or a concatenation of both can be used. The encoding is performed through word2vec [22] embedding in a 300-dimensional space. It is based on a Gensim model [30] pretrained on Google News dataset. For each word composing the textual input, a word2vec embedding is extracted. After that, they are aggregated by mean and unit-normalized, resulting in a vector for each image.

Multimodal features. To perform multimodal VSD, textual and visual features are combined. In [12], beyond the vector concatenation, Canonical Correlation Analysis and Deep Canonical Correlation Analysis are also explored. Nevertheless, their performances were poorer than concatenation ones, hence we explored only this last option.

3.2. Graph Construction

The core of our method relies on a graph-based semi-supervised learning algorithm, named *Graph Transduction Games* (GTG) [11]. Such method requires as input a weighted graph G, in which a set of labeled L and unlabeled nodes U are present, and a stochastic initial assignment X between nodes to labels (senses). The output is then a refined assignment matrix X which is the results of nodes interaction.

After extracting the desired embedding (visual, textual or multimodal), we construct a weighted graph $G = (V, E, \omega)$ with no self-loop over all the items in the dataset. Here V corresponds to all the pair <image, verb> in both set L and U, hence $V = L \cup U$. The set of edges $E \subseteq V \times V$ connects all the nodes and the function $\omega : e \in E \to \mathbb{R}_{\geq 0}$ weighs the pairwise similarity between vertices.

We define the similarity ω between node i and j (the edges weight), as the cosine[4] of their d-dimensional features embedding f_i and f_j:

$$\omega_{i,j} = \begin{cases} \sum_{i=1}^{d} f_{i,d} \cdot f_{j,d} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

Within this context, f_i is computed considering one of the modalities presented above. In the experimental section we report performances for all the embeddings (mono-modality) and their combinations (multi-modality). All the pairwise similarities $\omega_{i,j}$ are stored in a matrix $W \in \mathbb{R}^{n \times n}$.

3.3. Initial assignment

The goal of the transductive process is to propagate the labels from the labeled set L to the unlabeled ones U. For this purpose each node $i \in V$ is paired with a probability vector x_i over the possible senses ($x_i \in \Delta^m$ where m is the number of senses and Δ^m is standard m-dimensional simplex). Such vector is initialized in two different ways, based on fact that it belongs to a labeled or an unlabeled node. For the labeled node:

$$x_{i,h} = \begin{cases} 1 & \text{if } i \text{ have sense } h \\ 0 & \text{otherwise} \end{cases}$$

while for the unlabeled nodes:

$$x_{i,h} = \begin{cases} \frac{1}{|S_i|} & \text{if } h \in S_i \\ 0 & \text{otherwise} \end{cases}$$

[2] We used the PyTorch implementation of VGG

[3] `https://github.com/karpathy/neuraltalk2` [15]

[4] Since the features are all non-negative with unit norm, the cosine can be computed using the dot product.

where $x_{i,h}$ corresponds to the probability that the i-th node chooses the label h, while S_i is the set of possible senses associated with the verb in the i-th node. All the assignment x_i with $i = \{1 \dots n\}$ are stored into a stochastic matrix $X \in \mathbb{R}^{n \times m}$.

3.4. Transductive Inference

The transductive inference is performed with a dynamical system, which is responsible to iteratively refine the initial assignment X. We define here two quantities:

$$u_{i,h} = \sum_{j \in U} (A_{ij} x_j)_h + \sum_{k=1}^{m} \sum_{j \in L_k} A_{ij}(h,k) \tag{1}$$

$$u_i = \sum_{j \in U} x_i^T A_{ij} x_j + \sum_{k=1}^{m} \sum_{j \in L_k} x_i^T (A_{ij})_k \tag{2}$$

where L_k is the set of nodes labeled with class k. The matrix $A_{ij} \in \mathbb{R}^{m \times m}$, is defined as $A_{ij} = I_m \times \omega_{ij}$ with I_m being the identity matrix and ω_{ij} the similarity of nodes i and j. The equation $u_{i,h}$ quantifies the support provided by the other nodes to the labeling hypothesis h for the node i. While the equation u_i quantifies the overall support received to the node i by the other nodes.

In the following, we add the time component t to distinguish between different iterative steps. For example, $x_i^{(t)}$ refers to the probability vector x_i at time t. The dynamical system, responsible for the assignment refinement, is formulated as follow:

$$x_{i,h}^{(t+1)} = x_{i,h}^{(t)} \frac{u_{i,h}^{(t)}}{u_i^{(t)}} \tag{3}$$

The Eq. 3 is repeated until all the vectors x_i stabilize. Such dynamical system is known as *replicator dynamics* [20] and mimics a natural selection process in which better-than-average hypothesis get promoted while others get extinct. It is worth noting that the refinement takes into account all the hypotheses of all the nodes. In this sense, the labeling is performed not in isolation but is the result of nodes interactions. The rationale is that similar nodes tend to have the same label. The more two nodes are similar, the more they will affect each other in picking the same class.

The Eq. 3 grants that the matrix X at convergence is a *labeling consistent* solution [14][28]. A weighted labeling assignment X is said to be consistent if:

$$\sum_{h=1}^{m} x_{i,h} u_{i,h} \geq \sum_{h=1}^{m} y_{i,h} u_{i,h} \; \forall i = 1, \dots, n$$

for all Y. This means that no other solution Y can perform better than X.

Finally, it is worth noted that Eq. 3 can be written in a matricial form for a fast GPU implementation:

$$x_i(t+1) = \frac{x_i(t) \odot (Wx(t))_i}{x_i(t)(Wx(t))_i^T} \tag{4}$$

where $\odot$ represents the Hadamard (element-wise) product.

Regarding the Eq.3, 10 iterations are typically sufficient to reach convergence [9].

4. Experiments

In this section, we reported the performances and the experimental settings of our proposed model, TVVSD. The experiments have been carried out on the only available benchmarks for this task, the VerSe and VerSe-19verbs datasets, following the same evaluation protocol as [12].

4.1. Datasets

The VerSe dataset [12] is composed of images selected from Common Objects in Context (COCO) [6] and Trento Universal Human Object Interaction (TUHOI) [16], 90 verbs and 163 possible senses, resulting in 3510 (image, verb) pairs. Verbs have been categorized as *motion* and *non-motion* based on Levin verb classes [18], resulting in 39 motion and 51 non-motion verbs.

Further, we reported performances on a subset of VerSe, named *VerSe-19verbs*, which is composed of verbs that have at least 20 images and at least two senses in the VerSe dataset, resulting in 19 motion and 19 non-motion verbs.

4.2. Competitors and baselines

We compare our method with two state-of-the-art algorithms: Gella et al. [12] and Silberer et al. [33]. To the best of our knowledge [12] and [33] are the only literature works dealing with the research problem of VVSD and reported performances on the VerSe dataset. The work of [12] is based on a variant of Lesk algorithm [17] in which the sense is assigned based on the cosine similarity between the embedding of <image, verb> and the possible verb-senses in the dictionary. This procedure does not require labeled data since the final choice is based on the maximum score between <image, verb > and all the possible senses associated to the same verb as the image. For this reason, we tagged this method as *unsupervised*.

[12] proposed a supervised setting, in which a logistic regression is trained on each different embeddings. Similarly to [12], in [33] a logistic regression is trained, but it uses a frame-semantic image representation of the <image, verb> pairs rather than the embedding generated by [12]. Both methods are categorized as *supervised* and performances are reported only on the VerSe-19verb dataset.

The performances of the first sense (FS) and most frequent sense (MFS) heuristics are shown in table 1 and 2. Both are widely used in the NLP literature [24] and considered respectively as baselines for unsupervised and supervised scenarios in [12]. The FS corresponds to the first sense of a verb in the dictionary representing the common sense in a language (not in a specific dataset), while MFS is the most frequent sense in a dataset. MFS is considered as a supervised heuristic since all labeled data are needed to compute it.

Using GOLD annotations for objects and captions										
				Textual			Visual	Concat (CNN+)		
	Images	FS*	MFS*	O	C	O+C	CNN	O	C	O+C
Motion - Unsupervised Gella et al. [12]	1812	70.8	86.2	54.6	73.3	75.6	58.3	66.6	74.7	73.8
Motion - TVVSD (1 lab/sense)	1812	70.8	86.2	**73.3±4.4**	73.4±6.1	74.2±5.5	**73.3±5.9**	**74.7±3.4**	74.6±5.3	74.1 ± 5.5
Motion - TVVSD (2 lab/sense)	1812	70.8	86.2	**79.4±4.4**	**83.1±2.7**	**83.3±2.6**	**78.8±4.6**	**80.7±4.5**	**84.0±2.9**	**83.6±3.2**
Motion - TVVSD (20 lab/sense)	1812	70.8	86.2	**97.1±0.8**	**92.8±0.06**	**92.8±0.06**	**95.9±0.06**	**96.8±1.0**	**92.8±0.05**	**92.9±0.1**
NonMotion - Gella et al. [12]	1698	80.6	90.7	57.0	72.7	72.6	56.1	66.0	72.2	71.3
NonMotion - TVVSD (1 lab/sense)	1698	80.6	90.7	**71.7±4.7**	71.3±4.4	75.8±4.0	**64.4±6.6**	**71.4±4.9**	70.8 ±4.7	74.8±4.0
NonMotion - TVVSD (2 lab/sense)	1698	80.6	90.7	**82.5±3.2**	**81.8±2.9**	**82.2±4.0**	**80.8±4.0**	**83.4±2.8**	**81.8±2.0**	**81.7±3.0**
NonMotion - TVVSD (20 lab/sense)	1812	80.6	90.7	**92.0 ±0.8**	**91.8±0.7**	**91.6±0.4**	**94.0 ±2.2**	**92.1±1.3**	**92.0±1.1**	**91.9±1.0**
Using PRED annotations for objects and captions										
				Textual			Visual	Concat (CNN+)		
	Images	FS*	MFS*	O	C	O+C	CNN	O	C	O+C
Motion - Unsupervised Gella et al. [12]	1812	70.8	86.2	65.1	54.9	61.6	58.3	72.6	63.6	66.5
Motion - TVVSD (1 lab/sense)	1812	70.8	86.2	**71.2±6.4**	**71.5±3.9**	**71.9±5.1**	**73.3±5.9**	73.0±6.3	**73.8±4.3**	**74.0±3.6**
Motion - TVVSD (2 lab/sense)	1812	70.8	86.2	**79.5±4.9**	**77.1±3.9**	**80.2±4.4**	**78.7±4.6**	**80.2±4.4**	**77.7±3.6**	**78.3±3.7**
Motion - TVVSD (20 lab/sense)	1812	70.8	86.2	**94.4±0.4**	**92.7±0.1**	**92.8±0.2**	**95.9±0.6**	**94.1±0.5**	**92.9±0.2**	**92.9±0.3**
NonMotion - Gella et al. [12]	1698	80.6	90.7	59.0	64.3	64.0	56.1	63.8	66.3	66.1
NonMotion - TVVSD (1 lab/sense)	1698	80.6	90.7	**64.4±5.2**	**72.2±5.4**	**73.3±4.4**	**64.4±6.6**	65.6±6.0	**73.3±4.9**	**73.3±4.6**
NonMotion - TVVSD (2 lab/sense)	1698	80.6	90.7	**75.3±4.1**	**84.3±3.3**	**77.3±3.4**	**80.8±4.0**	**77.3±3.4**	**84.3±3.7**	**83.1±3.7**
NonMotion - TVVSD (20 lab/sense)	1698	80.6	90.7	**93.2±1.4**	**92.8±2.0**	**92.4±1.7**	**95.9±0.6**	**93.0±1.7**	**92.7±2.0**	**92.6±2.0**

Table 1: Accuracy scores on VerSe dataset using different sense and image representations. The bolds are relative to the performances of [12]. * FS and MFS can be considered as unsupervised and supervised references respectively.

4.3. Evaluation protocol and metric

We used the same evaluation metric as the competitors, that is the predicted sense accuracy. Being our setting semi-supervised, the accuracy is computed only on the unlabeled part leaving aside the labeled set. The accuracy is assessed considering two forms of textual annotations [12] for object labels and descriptions: *GOLD* and *PRED* (see Sec. 3.1 - Textual features).

4.4. Textual and visual representation features.

Considering the different features (see Sec 3.1) and their combinations there are 7 possible setups for the experiments, which are in line with [12]: captions (C), object labels (O), captions with object labels (C+O), CNN features (CNN), CNN features concatenated to captions (CNN+C), CNN features concatenated to object labels (CNN+O) and CNN features concatenated to captions with object labels (CNN+O+C).

4.5. Experimental setting

Here we describe our experimental setting to make the reported results reproducible. Being our model semi-supervised, we carried out experiments considering an increasing number of labeled samples, from 1 up to 20.

The labeled set is generated by random sampling the dataset. To adhere with the evaluation protocol used by our competitors, the sampling is performed differently whether the dataset used is VerSe or VerSe-19verbs.

VerSe setup: For VerSe, we perform sampling per *sense*. Since we do not need to tune any parameter in our method, the remaining samples constitute the unlabeled set where we compute the accuracy scores. For the experiments on the VerSe dataset, we sampled up to $n = 20$ elements per class since our performances were converging afterward.

VerSe-19verbs setup: For this dataset, the sampling is performed per *verb* [12]. To adhere to the competitors [12, 33] we used the same split ratios $80/10/10^5$ for train/val/test but, since we don't have a real training phase, our labeled set is composed by the 80% of image-verb and the remaining part (20%) becomes the unlabeled set. We experimented up to sampling 80% of the minimal verb class which contains 20 images, thus experimented up to $n = 16$ labeled samples per verb class. The points which are not sampled are considered as unlabeled and the final accuracy is computed accordingly.

To account for data variability in the sampling process, we performed the experiments 15 times using different random-seeds and reported means and standard deviation.

5. Results

Here we report the results of our experiments and the ablation studies to assess parameter sensitivity. In particular, we consider the behavior of our model when the *labeled sample per class* (*lpct*) increases.

5.1. Performance evaluation on VerSe

We present the accuracy scores for TVVSD on VerSe dataset in Table 1. We reported our experimental results when one, two and 20 *lpc* are used. The performances of the intermediate amount of labeled elements (3 to 19) are reported in the ablation study (see Fig.2 and Fig.3). We highlighted our results in bold when our performances are better than [12] and the performances of [12] are not in the standard deviation range of our model.

TVVSD using one labeled sample per class: As a first step, we investigate the performances of our model considering 1 *lpc*. Despite being an extreme case, TVVSD outperformed the unsupervised model of [12] on 3 different features over 7 in both motion and non-motion verb classes (see Table 1). Considering the two heuristics (FS and MFS),

[5] The authors of [12] sent us these quantities since they were not specified in the paper.

Using GOLD annotations for objects and captions									
			Textual			Visual	Concat (CNN+)		
	FS*	MFS*	O	C	O+C	CNN	O	C	O+C
Motion - Unsupervised Gella et al. [12]	60.0	76.1	35.3	53.8	55.3	58.4	48.4	66.9	58.4
Motion - our (1 lab/sense)	60.0	76.1	**62.3±7.4**	56.6±8.3	58.6±8.2	59.1±8.3	**64.7±5.8**	58.8±7.0	59.0±8.2
Motion - our (2 lab/sense)	60.0	76.1	**71.0±8.8**	**68.1±6.3**	**69.7±7.9**	**67.1±7.6**	**72.9±6.9**	**70.3±6.1**	**70.6±8.1**
Motion - our (16 lab/sense)	60.0	76.1	**90.2±3.9**	**88.5±0.5**	**88.8±0.1**	**90.7±0.7**	**90.0±3.5**	**88.7±1.3**	**88.8±0.1**
Motion - Supervised Gella et al. [12]	60.0	76.1	82.3	78.4	80.0	82.3	83.0	82.3	83.0
NonMotion - Unsupervised Gella et al. [12]	71.3	80.0	48.6	53.9	66.0	55.6	56.5	56.5	59.1
NonMotion - our (1 lab/sense)	71.3	80.0	56.6±13.8	54.3±8.5	59.9±7.1	46.3±9.1	57.1±11.8	51.9±7.1	55.6±8.2
NonMotion - our (2 lab/sense)	71.3	80.0	**69.4±8.9**	**71.6±5.7**	**71.5±6.7**	**69.2±4.3**	**69.4±8.8**	**71.7±5.5**	**71.8±6.8**
NonMotion - our (16 lab/sense)	71.3	80.0	**91.4±2.4**	**90.6±1.5**	**90.2±1.5**	**94.2±2.9**	**91.4±2.5**	**91.0±2.1**	**90.6±1.5**
NonMotion - Supervised Gella et al. [12]	71.3	80.0	79.1	79.1	79.1	80.0	80.0	80.0	80.0

Using PRED annotations for objects and captions									
			Textual			Visual	Concat (CNN+)		
	FS*	MFS*	O	C	O+C	CNN	O	C	O+C
Motion - Unsupervised Gella et al. [12]	60.0	76.1	43.8	41.5	45.3	58.4	60.0	53.0	55.3
Motion - our (1 lab/sense)	60.0	76.1	**57.3±5.7**	**55.2±8.1**	**56.1±7.4**	59.1±8.1	58.2±6.9	58.1±7.9	58.2±6.3
Motion - our (2 lab/sense)	60.0	76.1	**63.0±9.0**	**61.2±7.7**	**61.4±8.8**	**67.1±7.6**	64.9±8.0	**62.9±6.8**	**64.1±7.4**
Motion - our (16 lab/sense)	60.0	76.1	**86.9±4.1**	**87.3±0.2**	**87.4±0.2**	**90.6±0.7**	**87.6±2.4**	**87.8±0.7**	**87.6±0.3**
Motion - Supervised Gella et al. [12]	60.0	76.1	80.0	69.2	70.7	82.3	83.0	82.3	83.0
Motion - Supervised Silberer et al.$^{\#}$ [33]	71.3	80.0	-	-	-	-	84.8 ± 0.69	-	-
NonMotion - Gella et al. [12]	71.3	80.0	46.0	61.7	55.6	55.6	52.1	60.0	55.6
NonMotion - our (1 lab/sense)	71.3	80.0	52.4±10.5	55.8±9.1	55.8±9.0	46.3±9.2	53.5±8.4	55.1±8.1	54.9±7.4
NonMotion - our (2 lab/sense)	71.3	80.0	**61.7±5.6**	**75.4±4.2**	**75.6±4.2**	**69.2±4.3**	**63.6±3.4**	**76.0±3.4**	**74.5±6.2**
NonMotion - our (16 lab/sense)	71.3	80.0	**92.2±2.8**	**93.8±3.0**	**93.0±3.1**	**94.2±2.9**	**92.3±2.9**	**93.8±3.0**	**93.0±3.1**
NonMotion - Supervised Gella et al. [12]	71.3	80.0	78.2	77.3	77.3	80.0	80.0	80.3	80.0
NonMotion - Supervised Silberer et al.$^{\#}$ [33]	71.3	80.0	-	-	-	-	80.4 ± 0.57	-	-

Table 2: Sense prediction accuracy using *PRED* and *GOLD* settings in VerSe-19verbs for unsupervised, semisupervised and supervised approaches using different types of senses and image representation features In **bold** the results that outperform the supervised method, while in **blue** the ones outperforming the unsupervised model. $^{\#}$ uses a different embedding than [12], hence direct comparisons are not straightforward.

TVVSD performed on par with FS while is not able to reach the MFS performances. This confirms the nature of our model, being semi-supervised its performances are typically bounded between unsupervised (FS and [12]) and supervised (MFS) methods.

TVVSD using two labeled samples per class: When adding an extra labeled point, hence having 2 *lpc*, TVVSD outperforms the unsupervised state-of-the-art [12] and FS heuristic in all features modalities and classes (motion and non-motion) with a large margin. The MFS is still performing better, but considers all the labels.

TVVSD using 20 labeled sample per class: In this experiment, we moved to the other extreme in terms of annotated data, providing a lot of labeled samples to our model. The result is that TVVSD significantly outperforms both [12], FS and MFS heuristic in all features types for both motion and non-motion verbs and in both GOLD and PRED settings. This is a remarkable result, in particular because we outperformed MFS, which uses the entire labeled dataset.

Additional considerations: It is worth noting that the standard deviation of our experiments, when considering 1 *lpc* is very high and is getting smaller increasing the labeled set size. This is obvious since we are adding labeled informations to our model. Moreover, although multimodality provides strong performance gain in the PRED setting rather than in the GOLD setting, it brings only a marginal improvement in TVVSD compared to [12]. This might be explained by the nature of the GTG algorithm, which exploits all the possible relations between samples in the datasets, hence the unimodal features might be sufficient. In fact, in general, TVVSD gives better performances considering unimodal features.

5.2. Performance evaluation on VerSe-19verbs

In Table 2 we summarized our performances on the VerSe-19verbs for GOLD and PRED settings.

Using GOLD annotations for objects and captions: As in the VerSe experiment, we tested our model under different *lpc*, employing 1, 2 and 16 *lpc*. We started the experiments considering the extreme case in which we have only 1 *lpc*. In this case, TVVSD achieves quite low performances, outperforming the unsupervised model of [12] only in 2 cases (both considering the O features) and only in the motion class (see Table 2). Speculating, the motivation of these performances might be the following: motion verbs represent typically actions performed between specific entities/objects. For example, consider the verb ”play” associated to an image containing a person and a musical instrument. The association with the correct sense is straightforward due to the two entities. That’s why, in the case of motion verbs, the O’s feature has a strong influence on the overall performances.

The remaining results are comparable (considering the standard deviation) or slightly worst than the competitors. Regarding the heuristics, TVVSD reaches comparable performances to FS only in the motion class, while reaching absolutely unsatisfying results compared to MFS and in the entire non-motion verb class.

When we tested the model with 2 *lpc* we got something interesting. With just 2 *lpc*, we reached better performances than the unsupervised model in both motion and non-motion verb classes. Regarding the heuristics, we achieved comparable or better results than FS, while MFS is still the stronger competitor. It is worth noting that we used only 2 *lpc*, hence the annotation effort is dramatically low. As in the VerSe experiment we tested our model on the other extreme case, with 16 *lpc*. In this case, we strongly surpass both the heuristics, the unsupervised model and also the supervised ones (where at least 16 *lpc* are used in training). This is a remarkable results, considering that MFS relies on the entire labeled dataset.

Using PRED annotations for objects and captions: We performed the same experiments conducted in the GOLD setting but considering the PRED data. Differently from the previous experiment, when 1 *lpc* is considered, TVVSD clearly outperforms the unsupervised competitor in the motion setting in 3 over 7 cases and performed on par in the remaining. Regarding the non-motion setting, TVVSD is poorly performing and is not able to outperform both the heuristics and the unsupervised model in [12]. As in the GOLD setting, when using just 2 *lpc*, the performances start to increase considerably. In this case, TVVSD reaches better performances than the unsupervised model and performed on par or better than the FS heuristics in the motion setting. Considering the non-motion verbs, TVVSD outperformed completely the unsupervised model of [12] but is still under-performing with respect to MFS and the supervised model of [12].

When 16 *lpc* are used, TVVSD significantly outperforms both supervised [12] and the MFS heuristic in both motion and non-motion verbs settings.

Regarding the PRED setting, another work reported sense prediction performances on VerSe-19verbs. In [33] the authors used the same logistic classifier and evaluation protocol as in [12] but with a different feature embedding, called *ImgObjLoc*. Indeed, [33] outperformed [12] showing that their feature model is more expressive and powerful. Nevertheless, when considering 16 *lpc*, our model with standard features outperforms the [33]. The gap is large, we gain 3 points and more than 10 points in the motion and non motion settings respectively. We left as future work, applying TVVSD to the features of [33].

5.3. Ablation of TVVSD

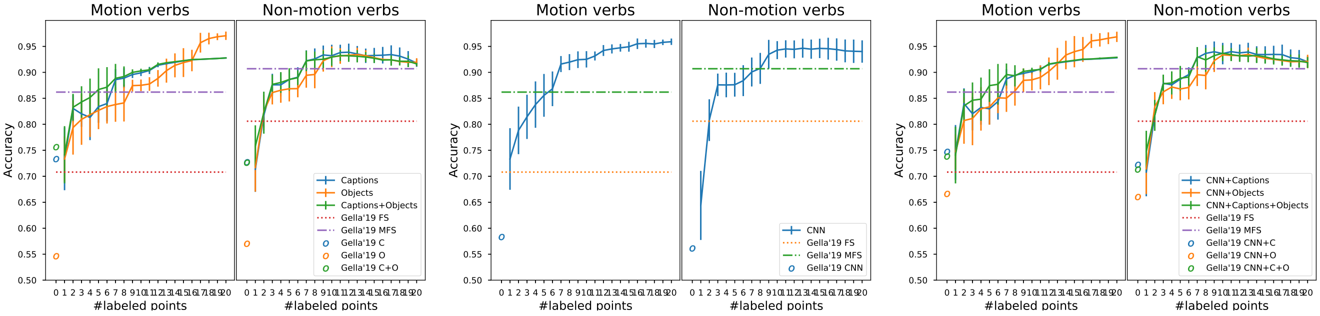
In this ablation study, we reported the performances of TVVSD when the labeled set size increases. This analysis is particularly useful to assess the effort needed for data annotation. The results on the VerSe dataset are shown in Figures 2 and 3, while for the VerSe-verb19 are in Figures 4 and 5. We reported means and standard deviations for all 15 runs. Alongside our results we added the performances of FS, MFS and Unsupervised [12] (see $lpc = 0$). The supervised results of [12] (see $lpc = 16$) are reported only for VerSe-verb19.

Ablation on VerSe dataset As can be seen, the performances with 1 labeled point per sense are comparable or better than [12] while with 2 or more labeled points we outperform the state-of-the-art [12]. This confirms that, for this task, few labeled points are sufficient, hence the labeling effort can be drastically reduced. After around 6 labeled points, we outperfom MFS significantly. In general, we noted that the higher the number of labeled points per sense, the greater the overall accuracy and smaller the standard deviation. The accuracy follows a logarithmic growth, i.e., the variation of the number of labels has a relevant role when they are few, whereas, with more than 6-8 labeled points per class, the accuracy starts converging. For the non-motion verbs, when textual features are used, the performance starts to decrease after reaching to a peak. This shows us that high number of labeled points creates a noise effect at these settings, i.e. similar elements in different classes mislead classification accuracy. We also noted that when visual features are used, the performance converges for the non-motion verbs while it continues to increase for the motion verbs. This was actually expected since actions of motion verbs are apparently more recognizable on images.

Ablation on VerSe-19verbs dataset Both figures 4 and 5 shown similar behaviors to the ablation on the VerSe dataset. We can drawn similar considerations, and noting that after 6-8 *lpc* the TVVSD model outperforms all the competitors showing a strong stability in the performances.

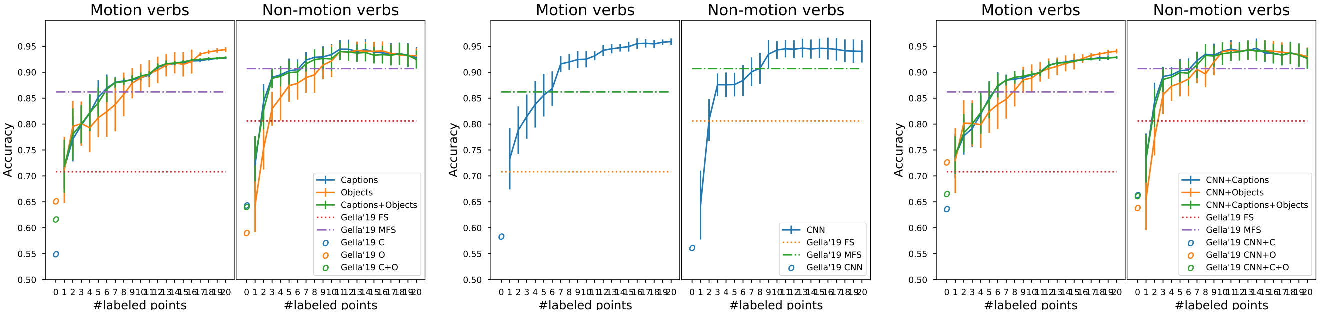
6. Conclusions

In this paper, we proposed a new model for multimodal VVSD tasks based on a transductive semi-supervised learning method. The proposed method is principled, well-funded and outperforms consistently the competitors. The transductive reasoning, used to perform the verb-sense disambiguation, considers the similarity of all the elements in the dataset, reaching a global consensus exploiting a *label consistency* principle. This differs completely from the (small) literature, which still relies on inductive methods that disambiguate visual verb in isolation. The proposed model is general enough to handle both unimodal and multimodal embeddings of <image,verb> pairs. Furthermore, we showed that 2 labeled points per sense are sufficient to outperform unsupervised state-of-the-art methods while 6-8 points are enough to obtain better performances than fully-supervised disambiguation models.



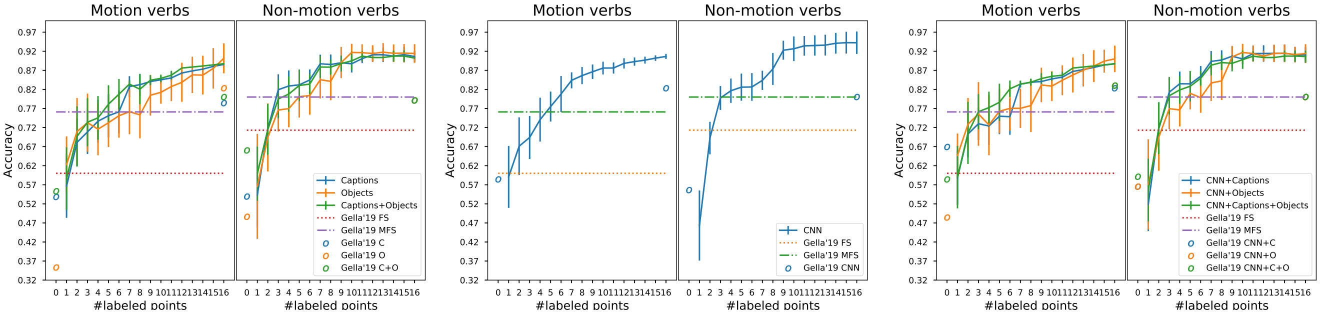
(a) text (b) cnn (c) cnn+text

Figure 2: GOLD results on VerSe for text data, cnn and cnn+text varying the number of labeled points in comparison with Gella et al. [12] approach (circles), FS and MFS results from [12] (dashed lines)



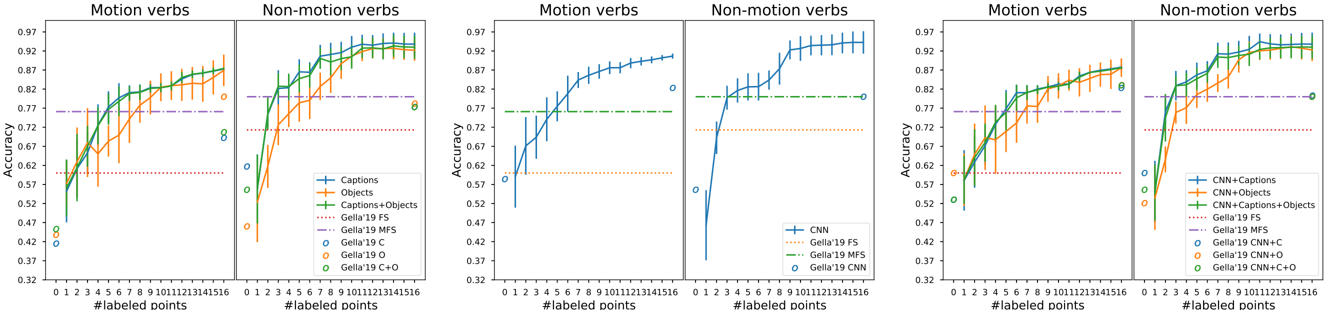
(a) text (b) cnn (c) cnn+text

Figure 3: PRED results on VerSe for text data, cnn and cnn+text varying the number of labeled points in comparison with Gella et al. [12] approach (circles), FS and MFS results from [12] (dashed lines). The central plot is repeated on purpose to avoid a blank figure.



(a) text (b) cnn (c) cnn+text

Figure 4: GOLD results on VerSe-verb19 for text data, cnn and cnn+text varying the number of labeled points in comparison with Gella et al. [12] approach (circles), FS and MFS results reported from [12] (dashed lines).



(a) text (b) cnn (c) cnn+text

Figure 5: PRED results on VerSe-verb19 for text data, cnn and cnn+text varying the number of labeled points in comparison with Gella et al. [12] approach (circles), FS and MFS results reported from [12] (dashed lines).

References

[1] Domenico Alfano, Roberto Abbruzzese, and Donato Cappetta. Neural semantic role labeling using verb sense disambiguation. In *CLiC-it*, 2019.

[2] Sinem Aslan, Sebastiano Vascon, and Marcello Pelillo. Two sides of the same coin: Improved ancient coin classification using graph transduction games. *Pattern Recognition Letters*, 131:158–165, 2020.

[3] Kobus Barnard and Matthew Johnson. Word sense disambiguation with pictures. *Artificial Intelligence*, 167(1-2):13–30, 2005.

[4] Yevgeni Berzak, Andrei Barbu, Daniel Harari, Boris Katz, and Shimon Ullman. Do you see what i mean? visual resolution of linguistic ambiguities. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1477–1487, Lisbon, Portugal, 2015. Association for Computational Linguistics.

[5] Teresa Botschen, Iryna Gurevych, Jan-Christoph Klie, Hatem Mousselly Sergieh, and Stefan Roth. Multimodal frame identification with multilingual evaluation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1481–1491, 2018.

[6] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[7] S. Cho and H. Foroosh. A temporal sequence learning for action recognition and prediction. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 352–361, 2018.

[8] Antonio Di Marco and Roberto Navigli. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(3):709–754, 2013.

[9] I. Elezi, A. Torcinovich, S. Vascon, and M. Pelillo. Transductive label augmentation for improved deep network learning. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1432–1437, Aug 2018.

[10] Ismail Elezi, Sebastiano Vascon, Alessandro Torcinovich, Marcello Pelillo, and Laura Leal-Taixé. The group loss for deep metric learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 277–294, Cham, 2020. Springer International Publishing.

[11] Aykut Erdem and Marcello Pelillo. Graph transduction as a noncooperative game. *Neural Computation*, 24(3):700–723, 2012.

[12] Spandana Gella, Frank Keller, and Mirella Lapata. Disambiguating visual verbs. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):311–322, 2019.

[13] Spandana Gella, Mirella Lapata, and Frank Keller. Unsupervised visual sense disambiguation for verbs using multimodal embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–192. Association for Computational Linguistics, 2016.

[14] Robert A Hummel and Steven W Zucker. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (3):267–287, 1983.

[15] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[16] Dieu-Thu Le, Jasper Uijlings, and Raffaella Bernardi. Tuhoi: Trento universal human object interaction dataset. In *Proceedings of the Third Workshop on Vision and Language*, pages 17–24, 2014.

[17] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. Citeseer, 1986.

[18] Beth Levin. *English verb classes and alternations: A preliminary investigation*. Univ. of Chicago press, 1993.

[19] Nicolas Loeff, Cecilia Ovesdotter Alm, and David A Forsyth. Discriminating image senses by clustering with multimodal features. In *Proceedings of the COLING/ACL 2006 main conference poster sessions*, pages 547–554, 2006.

[20] J. Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

[21] Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, 2016.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Conference Track Proceedings*, 2013.

[23] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[24] Roberto Navigli. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69, 2009.

[25] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics, 2010.

[26] Adrian Novischi and Dan Moldovan. Question answering with lexical chains propagating verb arguments. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 897–904, 2006.

[27] Tommaso Pasini and Roberto Navigli. Train-o-matic: Supervised word sense disambiguation with no (manual) effort. *Artificial Intelligence*, 279:103215, 2020.

[28] Marcello Pelillo. The dynamics of nonlinear relaxation labeling processes. *Journal of Mathematical Imaging and Vision*, 7(4):309–323, 1997.

[29] Sameer S Pradhan and Nianwen Xue. Ontonotes: The 90% solution. In *HLT-NAACL (Tutorial Abstracts)*, pages 11–12, 2009.

[30] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[31] Kate Saenko and Trevor Darrell. Unsupervised learning of visual sense models for polysemous words. In *Advances in Neural Information Processing Systems*, pages 1393–1400, 2009.

[32] K. Sharma, A. C. Kumar, and S. M. Bhandarkar. Action recognition in still images using word embeddings from natural language descriptions. In *2017 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 58–66, 2017.

[33] Carina Silberer and Manfred Pinkal. Grounding semantic roles in images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2616–2626, 2018.

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[35] Roman Sudarikov, Ondřej Dušek, Martin Holub, Ondřej Bojar, and Vincent Kríž. Verb sense disambiguation in machine translation. In *Proceedings of the Sixth Workshop on Hybrid Approaches to Translation (HyTra6)*, pages 42–50, 2016.

[36] Rocco Tripodi and Marcello Pelillo. A game-theoretic approach to word sense disambiguation. *Comput. Linguist.*, 43(1):31–70, Apr. 2017.

[37] Rocco Tripodi, Sebastiano Vascon, and Marcello Pelillo. Context aware nonnegative matrix factorization clustering. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1719–1724. IEEE, 2016.

[38] Sebastiano Vascon, Sinem Aslan, Alessandro Torcinovich, Twan van Laarhoven, Elena Marchiori, and Marcello Pelillo. Unsupervised domain adaptation using graph transduction games, 2019.

[39] Sebastiano Vascon, Marco Frasca, Rocco Tripodi, Giorgio Valentini, and Marcello Pelillo. Protein function prediction as a graph-transduction game. *Pattern Recognition Letters*, 2018.

[40] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[41] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, 2002.

[42] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003.