Weakly-Supervised Alignment of Video With Text

P. Bojanowski[2,*] R. Lagugie[2,†] E. Grave[3,‡] F. Bach[2,†] I. Laptev[2,*] J. Ponce[1,*] C. Schmid[2,§]
[1]ENS / PSL Research University [2]INRIA [3]Columbia University

Abstract

Suppose that we are given a set of videos, along with natural language descriptions in the form of multiple sentences (e.g., manual annotations, movie scripts, sport summaries etc.), and that these sentences appear in the same temporal order as their visual counterparts. We propose in this paper a method for aligning the two modalities, i.e., automatically providing a time stamp for every sentence. Given vectorial features for both video and text, we propose to cast this task as a temporal assignment problem, with an implicit linear mapping between the two feature modalities. We formulate this problem as an integer quadratic program, and solve its continuous convex relaxation using an efficient conditional gradient algorithm. Several rounding procedures are proposed to construct the final integer solution. After demonstrating significant improvements over the state of the art on the related task of aligning video with symbolic labels [7], we evaluate our method on a challenging dataset of videos with associated textual descriptions [36], using both bag-of-words and continuous representations for text.

1. Introduction

Fully supervised approaches to action categorization perform well for short video clips [44]. However, when the goal is not to classify a clip where a single action happens, but instead to compute the temporal extent of an action in a long video where multiple activities may take place, new difficulties arise. In fact, the task of identifying short clips where a single action occurs is at least as difficult as classifying the corresponding action afterwards. This is reminiscent of the gap in difficulty between categorization and detection in still images. In addition, as noted in [7], manual annotations are very expensive to get, even more so when working with a long video clip or a film shot, where many actions can occur. Finally, as mentioned in [13, 40], it is difficult to define exactly when an action occurs. This makes the task of understanding human activities much more difficult than finding objects or people in images.

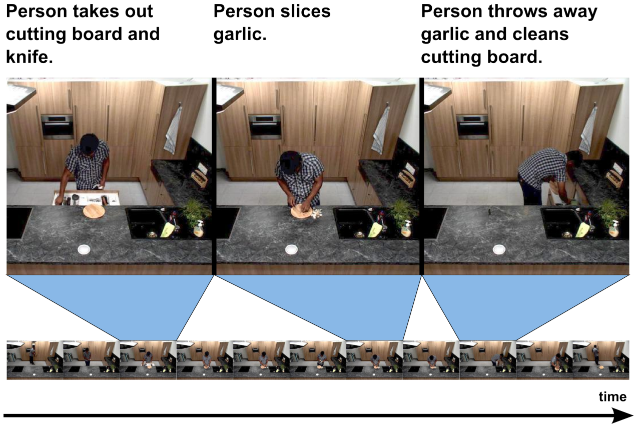

Person takes out cutting board and knife.
Person slices garlic.
Person throws away garlic and cleans cutting board.
time

Figure 1: An example of video to natural text alignment using our method on the TACoS [36] dataset.

In this paper, we propose to learn models of video content with minimal manual intervention, using natural language sentences as a weak form of supervision. This has the additional advantage of replacing purely symbolic and essentially meaningless hand-picked action labels with a semantic representation. Given vectorial features for both video and text, we address the problem of temporally aligning the video frames and the sentences (Figure 1) and assuming the order is preserved, with an implicit linear mapping between the two feature modalities. We formulate this problem as an integer quadratic program, and solve its continuous convex relaxation using an efficient conditional gradient algorithm. We discuss several rounding procedures to obtain a solution to the initial problem.

1.1. Related work

Many attempts at automatic image captioning have been proposed over the last decade: Duygulu *et al.* [9] were among the first to attack this problem, framing image recognition as a machine translation model. These ideas were further developed in [3]. A second important line of work builds simple natural language models as conditional random fields of a fixed size [10]. Typically this corre-

[*] WILLOW project-team, Département d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548, Paris, France.

[†] SIERRA project-team, Département d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548, Paris, France.

[‡] Department of Applied Physics & Applied Mathematics, Columbia University, New York, NY, USA.

[§] LEAR team, INRIA Grenoble Rhône-Alpes, Paris, France sponds to fixed language templates such as: ⟨Object, Action, Scene⟩. Much of the work on joint representations of text and images makes use of canonical correlation analysis (CCA) [19]. This approach has first been used to perform image retrieval based on text queries by Hardoon *et al.* [17], who learn a kernelized version of CCA to rank images given text. It has been extended to semi supervised scenarios [41], as well as to the multi-view setting [14]. All these methods frame the problem of image captioning as a retrieval task [18, 33]. Recently, there has also been an important amount of work on joint models for images and text using deep learning (*e.g.*, [12, 23, 28]).

There has been much less work on joint representations for text and video. A dataset of cooking videos with associated textual descriptions is used to learn joint representations of those two modalities in [36]. The problem of video description is framed as a machine translation problem in [37], while a deep model is proposed in [8]. Recently, a joint model of text, video and speech has also been proposed [29]. Textual data (such as scripts), has been used for automatic video understanding, for example for action recognition [26, 31]. Subtitles and scripts have also often been used to guide person recognition models (*e.g.*, [6, 35, 42]).

The temporal structure of videos and scripts has been used in several papers. In [7], an action label is associated with every temporal interval of the video while respecting the order given by an annotation sequence (see [35] for related work). The problem of aligning a large text corpus with video is addressed in [43]. The authors propose to match a book with its television adaptation by solving an alignment problem. The setting is however very different from ours, since the alignment is based on character identities only. The temporal ordering of actions, *e.g.*, in the form of Markov models or action grammars, has been used to constrain action prediction in videos [25, 27, 38]. Spatial and temporal constraints have also been used in the context of group activity recognition [1, 24].

The model we propose in this work is based on discriminative clustering, a weakly supervised framework for partitioning data. Contrary to standard clustering techniques, it uses a discriminative cost function [2, 16] and it has been used in image co-segmentation [20, 21], object co-localization [22], person identification in video [6, 35], and alignment of labels to videos [7].

Contrary to [7], for example, our work makes use of continuous text representations. Vectorial models for words are very convenient when working with heterogeneous data sources. Simple sentence representations such as bags of words are still frequently used [14]. More complex word and sentence representations can also be considered. Simple models trained on a huge corpus [32] have demonstrated their ability to encode useful information. It is also possible to use different embeddings, such as the posterior distribution over latent classes given by a hidden Markov model trained on the text [15].

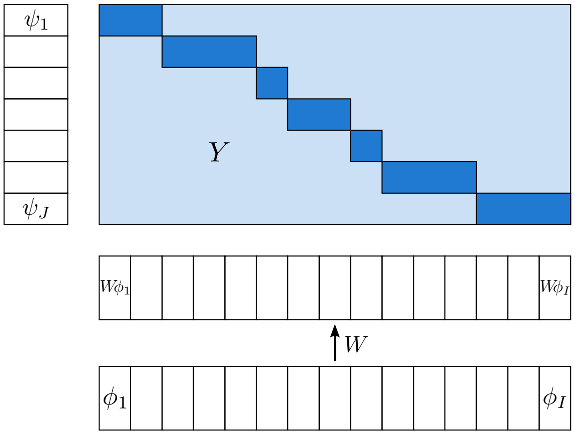


Figure 2: Illustration of some of the notations used in this paper. The video features Φ are mapped to the same space as text features using the map W. The temporal alignment of video and text features is encoded by the assignment matrix Y. Light blue entries in Y are zeros, dark blue entries are ones. See text for more details.

1.2. Problem statement and approach

Notation. Let us assume that we are given a data stream associated with two modalities, represented by the features $\Phi = [\phi_1, \dots, \phi_I]$ in $\mathbb{R}^{D \times I}$ and $\Psi = [\psi_1, \dots, \psi_J]$ in $\mathbb{R}^{E \times J}$. The elements of Φ (respectively Ψ) are indexed by i in $\{1, \dots, I\}$ (respectively j in $\{1, \dots, J\}$). Every element i is represented by a vector ϕ_i in $\mathbb{R}^D$. Similarly, every element j is represented by a vector ψ_j in $\mathbb{R}^E$.

In the context of video to text alignment, Φ is a description of the video signal, made up of I temporal intervals while Ψ is a textual description, composed of J sentences. However, our model is general and can be applied to other types of sequential data (biology, speech, music, *etc.*). In the rest of the paper, except of course in the experimental section, we stick to the abstract problem, considering two generic modalities of a data stream.

Problem statement. Our goal is to assign every element i in $\{1, \dots, I\}$ to exactly one element j in $\{1, \dots, J\}$. At the same time, we also want to learn a linear map[1] between the two feature spaces, parametrized by W in $\mathbb{R}^{E \times D}$. If the element i is assigned to an element j, we want to find W such that $\psi_j \approx W\phi_i$. If we encode the assignments as a binary matrix Y, this can be written in matricial form as follows: $\Psi Y \approx W\Phi$ (Fig. 2). The precise definition of the matrix Y will be provided in Sec. 2. Note that, in practice, we insert zero vectors in between the columns of Ψ. This allows some video frames not to be assigned to any text.

[1] As usual, we actually want an affine map. This can be done, by simply adding a constant row to Φ.

Relation with Bojanowski *et al.* [7]. Our model is an extension of [7] with several important improvements. In [7], instead of aligning video with natural language, the goal is to align video to symbolic labels in some predefined dictionary of size K ("open door", "sit down", *etc.*). By representing the labeling of the video using a matrix Z in $\{0,1\}^{K\times I}$, the problem solved there corresponds to finding W and Z such that: $Z \approx W\Phi$. The matrix Z encodes both data (which labels appear in each clip and which order) and the actual temporal assignments. Our parametrization allows us instead to separate the data features Ψ from the assignment variable Y. This has several significant advantages: first, this allows us to consider continuous text representations as the predicted output Ψ in $\mathbb{R}^{E\times J}$ instead of just classes. As shown in the sequel, this also allows us to easily impose natural, data-independent constraints on the assignment matrix Y.

1.3. Contributions

This article makes three main contributions: **(i)** we extend the model proposed in [7] in order to work with continuous representations of text instead of symbolic classes; **(ii)** we propose a simple method for including prior knowledge about the assignment into the model; and **(iii)** we demonstrate the performance of the proposed model on the task of aligning video with symbolic labels. We also exlore various feature representations on a challenging video dataset equipped with natural language meta data.

2. Proposed model

2.1. Basic model

Let us begin by defining the binary *assignment matrices* Y in $\{0,1\}^{J\times I}$. The entry Y_{ji} is equal to one if i is assigned to j and zero otherwise. Since every element i is assigned to exactly one element j, we have that $Y^T\mathbf{1}_J = \mathbf{1}_I$, where $\mathbf{1}_k$ represents the vector of ones in dimension k. As in [7], we assume that temporal ordering is preserved in the assignment. Therefore, if the element i is assigned to j, then $i+1$ can only be assigned to j or $j+1$. In the following, we will denote by $\mathcal{Y}$ the set of matrices Y that satisfy this property. This recursive definition, allows us to obtain an efficient dynamic programming algorithm for minimizing linear functions over $\mathcal{Y}$, which is a key step to the optimization method presented in Sec. 3.

We measure the discrepancy between ΨY and $W\Phi$ using the squared L_2 loss. Using an L_2 regularizer for the model W, our learning problem can be written as:

$$\min_{Y\in\mathcal{Y}} \min_{W\in\mathbb{R}^{E\times D}} \frac{1}{2I}\|\Psi Y - W\Phi\|_F^2 + \frac{\lambda}{2}\|W\|_F^2. \tag{1}$$

When Y is fixed, this is a ridge regression problem; In particular, we can rewrite (1) as:

$$\min_{Y\in\mathcal{Y}} q(Y), \tag{2}$$

where $q:\mathcal{Y}\to\mathbb{R}$ is defined for all Y in $\mathcal{Y}$ by:

$$q(Y) = \min_{W\in\mathbb{R}^{H\times D}} \left[\frac{1}{2I}\|\Psi Y - W\Phi\|_F^2 + \frac{\lambda}{2}\|W\|_F^2\right]. \tag{3}$$

For a fixed Y, the minimization with respect to W in (3) can be done in closed form, and its solution is:

$$W^* = \Psi Y \Phi^T\left(\Phi\Phi^T + I\lambda \mathrm{Id}_D\right)^{-1}, \tag{4}$$

where Id_k is the identity matrix in dimension k. Substituting in (3) yields:

$$q(Y) = \frac{1}{2I}\mathrm{Tr}\left(\Psi Y Q Y^T \Psi^T\right), \tag{5}$$

where Q is a matrix depending on the data and the regularization parameter λ:

$$Q = \mathrm{Id}_I - \Phi^T\left(\Phi\Phi^T + I\lambda\mathrm{Id}_D\right)^{-1}\Phi. \tag{6}$$

Multiple streams. Suppose now that we are given N data streams (video clips in our case), indexed by n in $\{1,\dots,N\}$. The approach proposed so far is easily generalized to this case by taking Ψ and Φ to be the horizontal concatenation of all the matrices Ψ_n and Φ_n. Also, the matrices Y in $\mathcal{Y}$ have to be block-diagonal in this case, where the diagonal blocks are the assignment matrices of every stream:

$$Y = \begin{bmatrix} Y_1 & & 0 \\ & \ddots & \\ 0 & & Y_N \end{bmatrix}.$$

This is the model actually used in our implementation.

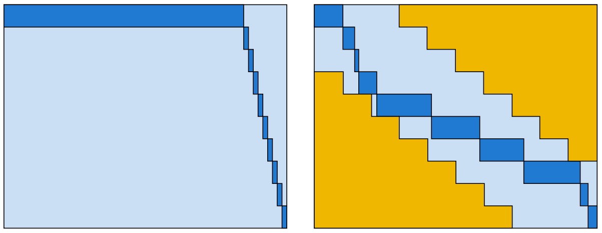
2.2. Priors and constraints

We can incorporate task-specific knowledge in our model by adding constraints on the matrix Y to model event duration for example. Constraints on Y can also be used to avoid the degenerate solutions known to plague discriminative clustering [2, 7, 16, 20].

Duration priors. The model as presented so far is solely based on a discriminative function. Our formulation in terms of an assignment variable Y allows us to reason about the number of elements i that get assigned to the element j. This in turn can be interpreted as the *duration* of the element j. More formally, the duration $\delta(j)$ of element j is obtained as:

$$\delta(j) = \mathbf{e}_j^T Y \mathbf{1}_I, \tag{7}$$

where $\mathbf{e}_j$ is the j-th vector of the canonical basis of $\mathbb{R}^J$. We penalize the squared deviation from a target duration μ.



(a) A degenerate solution. (b) A constrained solution.

Figure 3: **(a)** depicts a typical near degenerate solution where almost all the the elements i are assigned to the first element in $\{1, \dots, J\}$. This silution is close to the constant vector element of the kernel of Q. **(b)** To avoid this, we can force the alignment to stay outside of a given region (shown in yellow), which may be a band or a parallelogram. The dark blue entries correspond to the assignment matrix Y while the yellow ones represent the constraint set. See text for more details. (Best seen in color.)

Assuming for simplicity a single variance parameter for all units, this leads to the following duration penalty:

$$r(Y) = \frac{1}{2\sigma^2}\|Y\mathbf{1}_I - \mu\|_2^2. \tag{8}$$

Heuristics for estimating suitable values of σ and the mean vector μ will be discussed in Sec. 5. TODO

Path priors. Some elements in the set $\mathcal{Y}$ correspond to very unlikely assignments. In speech processing and various related tasks [34], the warping paths are often constrained, forcing for example the path to fall in the Sakoe-Chiba band or in the Itakura parallelogram [39]. Such constraints allow us to encode task-specific assumptions and to avoid degenerate solutions associated with the fact that constant vectors belong to the kernel of Q (Fig. 3 (a)). Band constraints, as illustrated in Fig. 3 (b), successfully exclude the kind of degenerate solutions presented in (a).

Let us denote by Y_c the band-diagonal matrix of width β, such that near diagonal entries are 0 and others are 1; such a matrix is illustrated in Fig. 3 (b) in yellow. In order to ensure that the assignment does not deviate too much from the diagonal, we can impose that at most C non zero entries of Y are outside the band. We can formulate that constraint as follows:

$$\text{Tr}(Y_c^T Y) \leq C.$$

This constraint could be added to the definition of the set $\mathcal{Y}$, but this would prohibit the use of dynamic programming, which is a key step to the optimization algorithm described in Sec. 3. We instead propose to add a penalization term to our cost function, corresponding to the Lagrange multiplier for this constraint. Indeed, for any value of C, there exists an α such that if we add

$$l(Y) = \alpha\text{Tr}(Y_c^T Y), \tag{9}$$

to our cost function, the two solutions are equal, and thus the constraint is satisfied. In practice, we select the value of α by grid search on a validation set.

2.3. Full problem formulation

Incorporating the constraints defined in Sec. 2.2 into our objective function yields the following optimization problem:

$$\min_{Y\in\mathcal{Y}}.\; q(Y) + r(Y) + l(Y), \tag{10}$$

where q, r and l are the three functions respectively defined in (5), (8) and (9).

3. Optimization

3.1. Continuous relaxation

The discrete optimization problem formulated in Eq. (2) is the minimization of a positive semi-definite quadratic function over a very large set $\mathcal{Y}$, composed of binary assignment matrices. It is NP hard, and following [7], we relax this problem by minimizing our objective function over the (continuous) convex hull $\overline{\mathcal{Y}}$ instead of $\mathcal{Y}$. Although it is possible to describe $\overline{\mathcal{Y}}$ in terms of linear inequalities, we never use this formulation in the following, since the use of a general linear programing solver does not exploit the structure of the problem. Instead, we consider the relaxed problem:

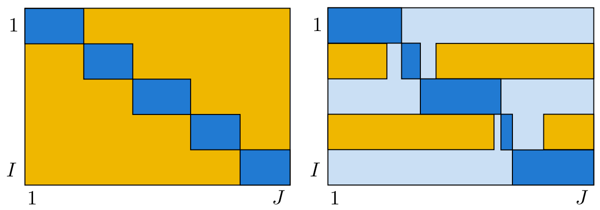
$$\min_{Y\in\overline{\mathcal{Y}}}.\; q(Y) + r(Y) + l(Y) \tag{11}$$

as the minimization of a convex quadratic function over an implicitly defined convex and compact domain. This type of problem can be solved efficiently using the Frank-Wolfe algorithm [7, 11] as soon as it is possible to minimize linear forms over the convex compact domain.

First, note that $\overline{\mathcal{Y}}$ is the convex hull of Y, and the solution to $\min._{Y\in\mathcal{Y}} \text{Tr}(AY)$ is also a solution of $\min._{Y\in\overline{\mathcal{Y}}} \text{Tr}(AY)$ ([5]). As noted in [7], it is possible to minimize any linear form $\text{Tr}(AY)$, where A is an arbitrary matrix, over $\mathcal{Y}$ using dynamic programming in two steps: First, we build the cumulative cost of matrix D whose entry (i, j) is the cost of the optimal alignment starting in $(1, 1)$ and terminating in (i, j). This step can be done recursively in $\mathcal{O}(IJ)$ steps. We then recover the optimal Y by backtracking in the matrix D. See [7] for details.

3.2. Rounding

Solving (11) provides a continuous solution Y^* in $\overline{\mathcal{Y}}$ and a corresponding optimal linear map W^*. Our original problem is defined on $\mathcal{Y}$, and we thus need to round Y^*. We propose three rounding procedures, two of them corresponding to Euclidean norm minimization problems and a third one using the map W^*. All three boil down to solving a linear problem over $\mathcal{Y}$ which can be solved, once again, using dynamic programming. Since there is no principled, analytical way to pick one of these procedures over the others,



(a) Y fixed to ground truth. (b) Corresponding constraints.

Figure 4: Two ways of incorporating supervision: **(a)** the assignments are fixed to the ground truth; the dark blue entries exactly correspond to Y_s **(b)** the assignments are constrained. For odd rows, we force the assignments to be outside of the golden strips.

we conduct an empirical evaluation in Sec. 5 to assess the strengths and weaknesses of each of them.

Rounding in $\mathcal{Y}$. The simplest way to round Y^* is to find the closest point Y according to the Euclidean distance in the space $\mathcal{Y}$:

$$\min_{Y \in \mathcal{Y}} \|Y - Y^*\|_F^2. \tag{12}$$

This problem is easily shown to reduce to a linear program over $\mathcal{Y}$.

Rounding using data. It is also possible to do the rounding in the space $\Psi\mathcal{Y}$. This is in fact the space where the original least-squares minimization is formulated. We solve in this case the problem

$$\min_{Y \in \mathcal{Y}} \|\Psi(Y - Y^*)\|_F^2, \tag{13}$$

which weighs the error measure using the feature Ψ. A simple calculation shows that Eq. (13) is equivalent to:

$$\min_{Y \in \mathcal{Y}} \operatorname{Tr}\left(Y^T \left(\mathbf{1}_I \operatorname{Diag}(\Psi^T\Psi)^T - 2\Psi^T\Psi Y^*\right)\right). \tag{14}$$

Rounding using W^*. Our optimization procedure gives us two outputs, namely a relaxed assignment $Y^* \in \overline{\mathcal{Y}}$ and a model W^* in $\mathbb{R}^{E \times D}$. We can use this model to predict an alignment Y in $\mathcal{Y}$ by solving the following quadratic optimization problem

$$\min_{Y \in \mathcal{Y}} \|\Psi Y - W^*\Phi\|_F^2.$$

As before, this is equivalent to a linear program. An important feature of this rounding procedure is that it can also be used on previously unseen data.

4. Semi-supervised setting

The proposed model is suited to semi-supervised learning. Incorporating additional supervision just consists in constraining parts of the matrix Y. Let us assume that we are given a triplet (Ψ_s, Φ_s, Y_s) representing supervisory data. The part of data that is not involved in that supervision is denoted by (Ψ_u, Φ_u, Y_u). Using the additional data amounts to solving (10) with matrices (Ψ, Φ, Y) defined as:

$$\Psi = [\Psi_u,\ \kappa\,\Psi_s], \Phi = [\Phi_u,\ \kappa\,\Phi_s], Y = \begin{bmatrix} Y_u & 0 \\ 0 & Y_s \end{bmatrix}. \tag{15}$$

The parameter κ allows us to weight properly the supervised and unsupervised examples. Scaling the features this way corresponds to using the following loss:

$$\|\Psi_u Y_u - W\Phi_u\|_F^2 + \kappa^2 \|\Psi_s Y_s - W\Phi_s\|_F^2. \tag{16}$$

Since Y_s is given, we can optimize over $\mathcal{Y}$ while constraining the lower right block of Y. From an implementation point of view, this corresponds to having the entries in Y_s fixed to the ground-truth values during optimization.

We propose to improve over this simple scheme in the following way: instead of fixing the entries of Y_s to the ground-truth assignment, we just constrain them. For odd (non null) elements j, we force the set of elements i that are assigned to j to be a subset of those observed in the ground truth. That way, we allow the assignment to pick the most discriminative parts of the video within the annotated interval. We illustrate these constraints in Fig. 4. This way of incorporating supervision yields empirically much better performance than using hard constraints.

5. Experimental evaluation

We evaluate the proposed approach on two challenging datasets. We first compare it to a recent method limited to symbolic labels on the associated dataset [7]. We then run several experiments on a video dataset composed of cooking activities with textual annotations called TACoS [36]. All results are reported with performance averaged over several random splits.

Performance measure. All experiments are evaluated using the *Jaccard measure* in [7], that quantifies the difference between a ground-truth assignment Y_{gt} and the predicted Y by computing the precision for each row. In particular the best performance of 1 is obtained if the predicted assignment is within the ground truth. If the prediction is outside, it is equal to 0.

5.1. Comparison with Bojanowski et al. [7]

Our model is a generalization of Bojanowski *et al.* [7]. Indeed, we can easily cast the problem formulated in that paper into our framework. Our model is different in three crucial ways: First, we do not need to add a separate "background class", which is always problematic. Second, we handle the semi-supervised setting differently, using the method described in Sec. 4 instead of the hard constraints in [7]. Most importantly, we replace the matrix Z by ΨY, allowing us to add data-independent constraints and priors on Y. In this section we describe comparative experiments conducted on the dataset proposed in [7].

Dataset. We use the video clips, labels and features provided by the authors of [7]. This data is composed of 843 video clips (94 clips are set aside) that are annotated with a sequence of labels. There are 16 different labels such as *e.g.* "Eat", "Open Door" and "Stand Up". As in [7], we randomly split the dataset into ten different validation, evaluation and supervised sets.

Features. The label sequences provided as weak supervisory signal in [7] can be used as our features Ψ. We consider a language composed of sixteen words, where every word corresponds to a label. Then, the representation ψ_j of every element j is the indicator vector of the j-th label in the sequence. Since we do not model background, we simply interleave zero vectors in between meaningful elements. The matrix Φ corresponds to the video features provided with the paper's code. These features are 2000-dimensional bag-of-words vectors computed on the HOF channel.

Baselines. As our baseline, we run the online available code[2] from [7] for different fractions of annotated data, seeds and parameters.

Results. We present a plot of performance versus amount of supervised data in Fig. 5. We use the same evaluation metric as in [7]. First of all, when no supervision is available, our method works significantly better (no overlap between error bars). This may be due (1) to the fact that we do not model background as a separate class; and (2) to the use of the priors described in Sec. 2.2. As additional supervisory data becomes available, we observe a consistent improvement of more than 5% over [7].

The main interesting point is the fact that the drop at the beginning of the curve in [7] does not occur in our case. When no supervised data is available, the optimal Y^* solely depends on the video features Φ. When the fraction of annotated data increases, the optimal Y^* changes and also depends on the annotations. However, manual annotations are not necessarily coherent with the Y^* obtained without supervision. Our way of dealing with supervised data is less coercive and does not commit strictly to the annotated data.

Please note that the best performing rounding procedure is the one using the data ψ. Also, the performance of Bojanowski *et al.* is slightly different from the numbers reported in [7] since the parameter grids are different.

5.2. Results on the TACoS dataset [36]

We also evaluate our method on the TACoS dataset [36]. This allows us to evaluate it on a corpus including actual natural language sentences.

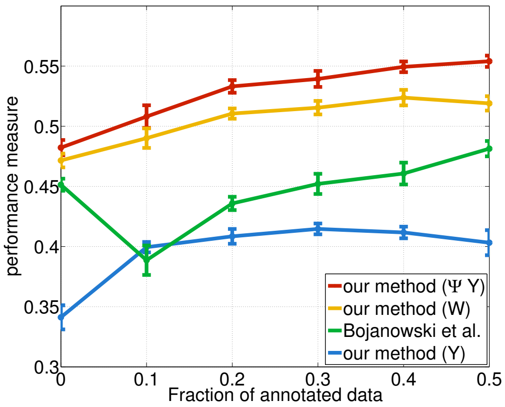


Figure 5: Comparing the new model to the model in [7] on the data from that paper.

Dataset. The TACoS dataset is composed of 127 videos picturing people who perform cooking tasks. Every video is associated with two kinds of annotations. The first one is composed of low-level activity labels with precise temporal location. We do not make use of these fine-grained annotations in this work. The second one is a set of natural language descriptions that were obtained by crowd-sourcing. Annotators were asked to describe the content of the video using simple sentences.

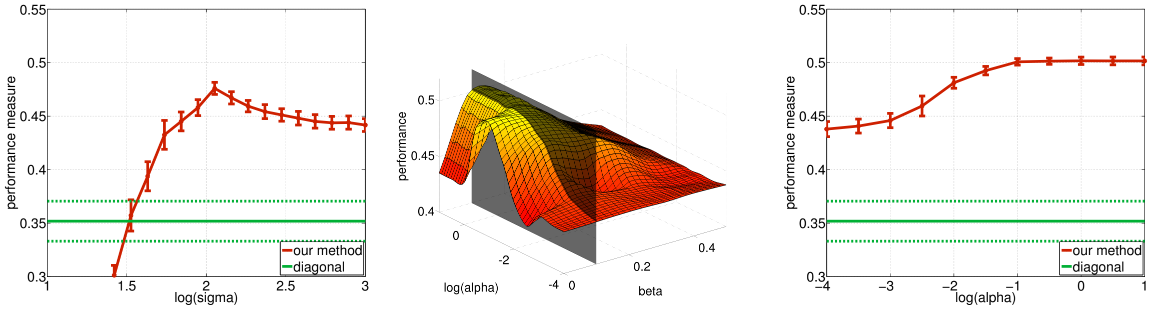
Each video Φ is associated with k textual descriptions $[\Psi^1, \dots, \Psi^K]$. Every textual description is composed of multiple sentences with associated temporal extent. We consider as data points the pairs (Ψ^k, Φ) for k in $\{1, \dots, K\}$. Doing so does not allow us to reason about the possible correlations between different descriptions of the same video.

Video features. We build the feature matrix Φ by computing dense trajectories [44] on all videos. We compute dictionaries of 500 words for HOG, HOF and MBH channels. For a given temporal window, we concatenate bag-of-words representations for the four channels. As in the Hellinger kernel, we use the square root of L_1 normalized histograms as our final features. We use temporal windows of length 150 frames with a stride of 50.

Text features. To apply our method to textual data, for every sentence in a textual description, we build a feature ψ_i. In our experiments, we explore multiple ways to represent sentences and empirically compare their performance (Table 1). We discuss two representations, one based on bag of words, the other on continuous word embeddings [32].

To build our bag-of-words representation, we construct a dictionary using all sentences in the TACoS dataset. We run a part-of-speech tagger and a dependency parser [30] in order to exploit grammatical structure. These features are pooled using three different schemes. (1) ROOT: In this setup, we simply encode each sentence by its root verb as

[2] https://github.com/piotr-bojanowski/action-ordering



(a) duration prior (b) band prior (c) band prior

Figure 6: Evaluation of the priors we propose in this paper. **(a)** We plot the performance of our model for various values of σ. When σ is big, the prior has no effect. We see that there is a clear trade off and an optimal choice of σ yields better performance. **(b)** Performance as a function of α and width of the band. The shown surface is interpolated to ease readability. **(c)** Performance for various values of α. This plot corresponds to the slice illustrated in (b) by the black plane.

provided by the dependency parser. (2) ROOT+DOBJ: In this setup we encode a sentence by its root verb and its direct object dependency. This representation makes sense on the TACoS dataset as sentences are in general pretty simple. For example, the sentence “The man slices the cucumber” is represented by “slice” and “cucumber”. (3) VNA: This representation is the closest to the usual bag-of-words text representation. We simply pool all the tokens whose part of speech is verb, noun or adjective. The two first representations are very rudimentary versions of bags of words. They typically contain only one or two non-zero elements.

We also explore the use of word embeddings [32], trained on three different corpora. First, we train them on the TACoS corpus. Even though the amount of data is very small (175,617 words), the vocabulary is also limited and the sentences are simple. Second, we train the vector representations on a dataset of 50,993 kitchen recipes, downloaded from allrecipes.com. This corresponds to a corpus of roughly 5 million tokens. However, the sentences are written in imperative mode, which differs from the sentences found in TACoS. For completeness, we also use the WaCky corpus [4], a large web-crawled dataset of news articles. We train representations of dimension 50, 100 and 200. A sentence is represented by the concatenation of the vector representations of its root verb and its root’s direct object.

Baselines. On this dataset, we consider two baselines. The first one is Bojanowski *et al.* [7] using the ROOT textual features. Verbs are used in place of labels by the method. The second one, that we call Diagonal, corresponds to the performance obtained by the uniform alignment. This corresponds to assigning the same amount of video elements i to each textual element j.

Evaluation of the priors. We propose in Sec. 2.2 two priors for including prior knowledge and avoiding degenerate solutions to our problem. In this section, we evaluate the performance of the two proposed priors on TACoS. To this end, we run our method with the two different models separately. We perform this experiment using the ROOT+DOBJ text representation. The results of this experiment are illustrated in Fig. 6.

We see that both priors are useful. The duration prior, when σ is carefully chosen, allows us to improve performance from 0.441 to 0.475. There is a clear trade-off in this parameter. Using a bit of duration prior helps us to get a meaningful Y^* by discarding degenerate solutions. However, when the prior is too strong, we obtain a degenerate solution with decreased performance.

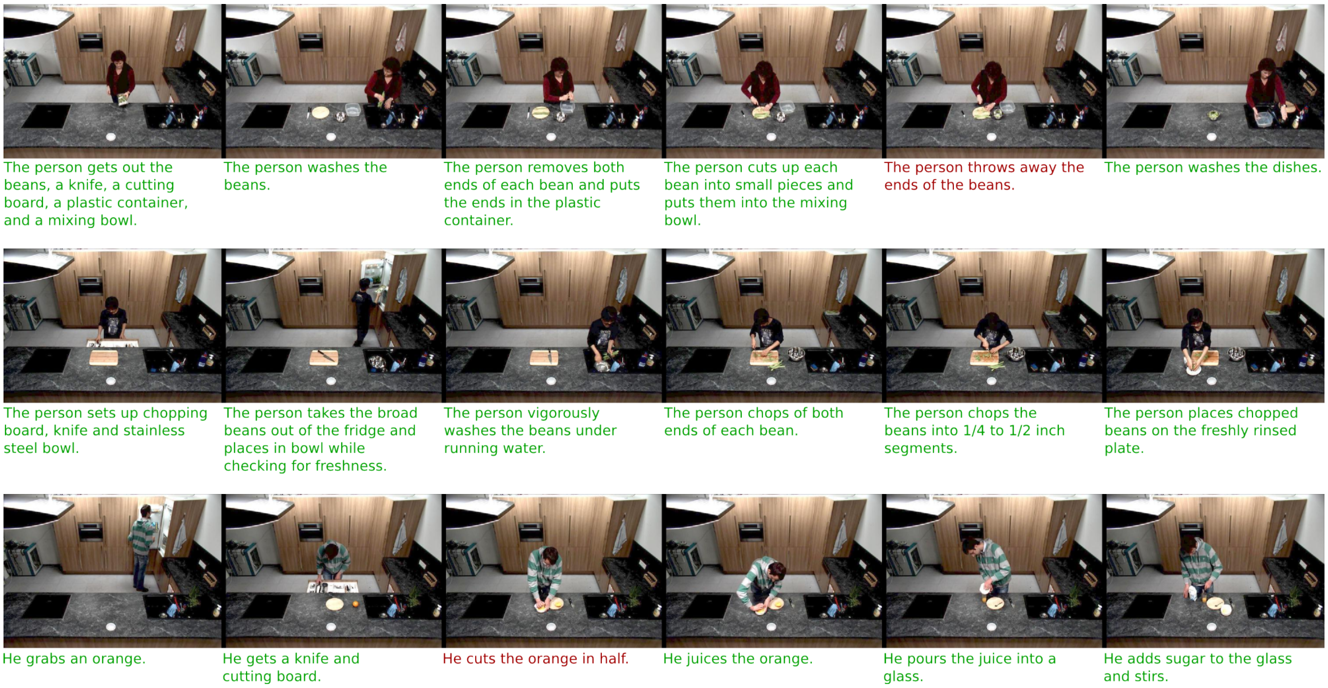
The band prior (as depicted in Fig. 6, b and c) improves

text representation	Dim. 100	Dim. 200
W2V UKWAC 100	43.8 (1.5)	46.4 (0.7)
W2V TACoS 100	48.3 (0.4)	48.2 (0.4)
W2V ALLRECIPE 100	43.3 (0.7)	44.7 (0.5)

Table 1: Comparison of text representations trained on different corpora, in dimension 100 and 200. The numbers in parenthesis correspond to the standard error between different splits.

text representation	nosup	semisup
Diagonal	35.2 (3.7)	
Bojanowski et al. [7]	39.0 (1.0)	49.1 (0.7)
ROOT	49.9 (0.2)	59.2 (1.0)
ROOT+DOBJ	48.7 (0.9)	65.4 (1.0)
VNA	45.7 (1.4)	59.9 (2.9)
W2V TACoS 100	48.3 (0.4)	60.2 (1.5)

Table 2: Performance when no supervision is available (nosup) and when half of the dataset is provided with time stamped sentences (semisup). The numbers in parenthesis correspond to the standard error between different splits.



The person gets out the beans, a knife, a cutting board, a plastic container, and a mixing bowl.

The person washes the beans.

The person removes both ends of each bean and puts the ends in the plastic container.

The person cuts up each bean into small pieces and puts them into the mixing bowl.

The person throws away the ends of the beans.

The person washes the dishes.

The person sets up chopping board, knife and stainless steel bowl.

The person takes the broad beans out of the fridge and places in bowl while checking for freshness.

The person vigorously washes the beans under running water.

The person chops of both ends of each bean.

The person chops the beans into 1/4 to 1/2 inch segments.

The person places chopped beans on the freshly rinsed plate.

He grabs an orange.

He gets a knife and cutting board.

He cuts the orange in half.

He juices the orange.

He pours the juice into a glass.

He adds sugar to the glass and stirs.

Figure 7: Representative qualitative results for our method applied on TACoS. Correctly assigned frames are in green, incorect ones in red.

performance even more. We plot in (b) the performance as a joint function of the parameter α and of the width of the band β. We see that the width that provides the best performance is 0.1. We plot in (c) the corresponding performance as a function of α. Using large values of α corresponds to constraining the path to be entirely inside the band, which explain why the performance flattens for large α. When using a small width, the best path is not entirely inside the band and one has to carefully choose the parameter α.

We show in Fig. 6 the performance of our method for various values of the parameters on the evaluation set. Please note however that when used in other experiments, the actual values of these parameters are chosen on a held-out validation set.

Evaluation of the text representations. We compare in Table. 1 the continuous word representations trained on various text corpora. Using the representation trained on TACoS works best. It is usually advised to retrain the representation on a text corpus that has similar distribution to the corpus of interest. Moreover, using higher-dimensional representations (200) does not help and can be explained by the limited size of the vocabulary. The features trained on a very large news corpus (UKWAC) benefit from using higher-dimensional vectors. With such a big corpus, the representations of the cooking lexical field are probably merged together.

In Table. 2, we experimentally compare our approach to the baselines, in an unsupervised setting and a semi-supervised one. First, we observe that the diagonal baseline has reasonable performance. Note that this diagonal assignment is different from a random one since a uniform assignment between text and video in our context makes some sense. Second, we compare our method to [7] on ROOT, which is the only set up where this method can be used. This baseline is higher than the diagonal one but pretty far from the performances of our model using ROOT as well.

When using bag-of-words representations, we notice that simple pooling schemes work best. Namely, the best performing representation is purely based on verbs. This is probably due to the fact that richer representations can mislead such a weakly supervised method. As additional supervision becomes available, the ROOT+DOBJ pooling works much better that only using ROOT, which validates this intuition.

6. Discussion.

We have presented in this paper a method for aligning a video with its natural language description. Our video representation is mainly based on dynamic features, mostly disregarding the crucial aspect of appearance. We would like to extend our work to even more challenging scenarios including feature movies and more complicated grammatical structures. Also, our use of natural language processing tools is limited, and we plan to incorporate better grammatical reasoning in future work.

Acknowledgements. This work was supported by a PhD fellowship from the EADS Foundation, the Institut Universitaire de France, the MSR-INRIA laboratory and ERC grants ALLEGRO, VideoWorld and Activia.

References

[1] M. R. Amer, S. Todorovic, A. Fern, and S.-C. Zhu. Monte carlo tree search for scheduling activity recognition. In *ICCV*, 2013.

[2] F. Bach and Z. Harchaoui. Diffrac: a discriminative and flexible framework for clustering. In *NIPS*, 2007.

[3] K. Barnard, P. Duygulu, D. A. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 2003.

[4] M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 2009.

[5] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[6] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding actors and actions in movies. In *ICCV*, 2013.

[7] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014.

[8] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*, 2014.

[9] P. Duygulu, K. Barnard, J. F. G. d. Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002.

[10] A. Farhadi, S. M. M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. A. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010.

[11] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 1956.

[12] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[13] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *CVPR*, 2011.

[14] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2014.

[15] E. Grave, G. Obozinski, and F. Bach. A markovian approach to distributional semantics with application to semantic compositionality. In *COLING*, 2014.

[16] Y. Guo and D. Schuurmans. Convex relaxations of latent variable training. In *NIPS*, 2007.

[17] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[18] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, pages 853–899, 2013.

[19] H. Hotelling. Relations between two sets of variates. *Biometrika*, 3:321–377, 1936.

[20] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[21] A. Joulin, F. Bach, and J. Ponce. Multi-class cosegmentation. In *CVPR*, 2012.

[22] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *ECCV*, 2014.

[23] A. Karpathy, A. Joulin, and F. F. F. Li. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in Neural Information Processing Systems*, pages 1889–1897, 2014.

[24] S. Khamis, V. I. Morariu, and L. S. Davis. Combining per-frame and per-track cues for multi-person action recognition. In *ECCV*, 2012.

[25] S. Kwak, B. Han, and J. H. Han. Scenario-based video event recognition by constraint flow. In *CVPR*, 2011.

[26] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[27] B. Laxton, J. Lim, and D. J. Kriegman. Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *CVPR*, 2007.

[28] R. Lebret, P. O. Pinheiro, and R. Collobert. Phrase-based image captioning. *arXiv preprint arXiv:1502.03671*, 2015.

[29] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *NAACL*, 2015.

[30] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014.

[31] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[32] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[33] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011.

[34] L. R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[35] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei. Linking people with "their" names using coreference resolution. In *ECCV*, 2014.

[36] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. Grounding action descriptions in videos. *TACL*, 1:25–36, 2013.

[37] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013.

[38] M. S. Ryoo and J. K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *CVPR*, 2006.

[39] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

[40] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *ECCV*, 2010.

[41] R. Socher and L. Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *CVPR*, 2010.

[42] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. "knock! knock! who is it?" probabilistic person identification in tv-series. In *CVPR*, 2012.

[43] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Book2movie: Aligning video scenes with book chapters. In *CVPR*, 2015.

[44] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.