Few-Shot Domain Adaptation for Named-Entity Recognition via Joint Constrained k-Means and Subspace Selection

Ayoub Hammal[1] **Benno Uthayasooriyar**[2,3] **Caio Corro**[4]
[1]Université Paris-Saclay, CNRS, LISN
[2]Data Analytics Solutions, SCOR [3]LMBA, CNRS, Université de Brest
[4]INSA Rennes, IRISA, Inria, CNRS, Université de Rennes

Abstract

Named-entity recognition (NER) is a task that typically requires large annotated datasets, which limits its applicability across domains with varying entity definitions. This paper addresses few-shot NER, aiming to transfer knowledge to new domains with minimal supervision. Unlike previous approaches that rely solely on limited annotated data, we propose a weakly supervised algorithm that combines small labeled datasets with large amounts of unlabeled data. Our method extends the k-means algorithm with label supervision, cluster size constraints and domain-specific discriminative subspace selection. This unified framework achieves state-of-the-art results in few-shot NER on several English datasets.

1 Introduction

Named-entity recognition (NER) is a fundamental information retrieval task that aims to identify entity mentions as well as their corresponding types in texts (Grishman and Sundheim, 1996; Chinchor and Robinson, 1998). This problem can be tackled via standard structured prediction methods, *e.g.* conditional random fields for segmentation (Lafferty et al., 2001; McCallum and Li, 2003; Sarawagi and Cohen, 2004). As supervised learning approaches come at the expense of building large annotated datasets, there is a growing interest in fine-tuning NER models using only a (very) small annotated dataset, called the support.

Specifically, we focus on the few-shot domain adaptation scenario: a model is first trained on a large annotated dataset, and then fine-tuned on target data using as support only 1-5 examples per mention type. We consider two different flavors of few-shot domain adaptation: (1) tag set extension, *i.e.* output domain transfer, where the model is fine-tuned to predict mention types that were previously unknown, but on the same input domain; (2) input domain transfer, where the model is fine-tuned to predict mentions in previously unseen data sources, potentially using a different annotation scheme. A natural approach in this setting is to build class prototypes from the support and rely on nearest neighbor classification for prediction (Fritzler et al., 2019; Yang and Katiyar, 2020; Das et al., 2022, *inter alia*).

In this work, we propose a novel weakly-supervised learning method for few-shot NER that overcome limitations of previous work. Firstly, our method is based on the k-means clustering algorithm which naturally benefits from access to extra unlabeled data that is often cheap and easy to collect. Secondly, we introduce a ratio constraint on the number of words that are not part of mentions as extra learning information to make the most of unlabeled data, in the same spirit as Effland and Collins (2021). To this end, we develop novel algorithms to take into account this ratio constraint in the training procedure. Thirdly and lastly, we jointly learn a projection of the data into a subspace so that clusters are well separated, often referred to as discriminative clustering (Ding and Li, 2007; Ye et al., 2007b). All in all, our procedure is grounded on a well-defined training problem and efficient optimization algorithms.

A well-known issue of few-shot learning is the instability of training, *i.e.* there can be a high variance between runs of the same training process for the same support, mainly due to source of randomness. To fix this issue, we devise a strictly deterministic training procedure, meaning that two runs will lead to the same results, as there not a single call to a random number generator. We achieve this by using batch updates (*i.e.* on the full training objective) instead of a stochastic optimization algorithm that operates on minibatches (Bottou, 2010). Moreover, we propose a deterministic initialization procedure for our k-means in order to bypass the usual random initialization. Finally, note that our training algorithm is based on a parameter-free opti-

Proceedings of the 31st International Conference on Computational Linguistics, pages 9902–9916
January 19–24, 2025. ©2025 Association for Computational Linguistics

mization method, meaning that there is no learning parameter to tune.

Our contributions can be summarized as follows:

- We formalize few-shot domain adaptation as a k-means clustering problem, which can benefit from extra unlabeled data;
- We propose novel algorithms for the E-step of k-means that allows to introduce ratio constraints in both hard and soft variants;
- We extend the clustering process to jointly project the data into a subspace where clusters are well-separated;
- We evaluate our approach in different few-shot settings and achieve novel state-of-the-art results compared to previous work.

Our code is publicly available.[1]

Notation. We write scalars (resp. sets) in lowercase (resp. uppercase), and vectors (resp. matrices) in bold lowercase (resp. uppercase). Given a matrix $\boldsymbol{M}$, we denote M_{ij} the element at row i and column j, and $\boldsymbol{M}_i$ the vector corresponding to row i. We denote $\|\boldsymbol{a}\| = \sqrt{\sum_i a_i^2}$ the L2 norm, $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{i,j} A_{ij} B_{ij}$ the sum of entries of the Hadamard product (*i.e.* dot product if arguments are vectors) and $\mathbf{tr}(\boldsymbol{M}) = \sum_i M_{ii}$ the trace. Given $i \in \mathbb{N}_{++}$, we write $[i]$ the set $\{1..i\}$. Given a set S, we write $\mathcal{P}(S)$ the powerset of S and $\mathcal{P}_i(S)$ the set of all subsets of S with cardinality i. We denote $\triangle(k) = \{\boldsymbol{a} \in \mathbb{R}^k_+ | \sum_i a_i = 1\}$ the simplex of dimension $k-1$.

2 Few-Shot Named-Entity Recognition

The NER problem aims to identify entity mentions in texts. This chunking task is often reduced to a word tagging problem using the BIO scheme (Ramshaw and Marcus, 1995): each word is tagged either with O (not in a mention), B-TYPE (first word of a mention) or I-TYPE (following words in a mention), where TYPE is any allowed mention type (*e.g.* LOC, ORG, etc.) Following previous work in the few-shot scenario (Yang and Katiyar, 2020; Das et al., 2022), we rely on a simplified IO scheme, where each word is either tagged with O or I-TYPE, for example:

U.N.	official	Ekeus	heads	for	Baghdad
I-ORG	O	I-PER	O	O	I-LOC

We write $T = \{$O, I-LOC, ...$\}$ the set of tags.

Weighting model. Let $\boldsymbol{s} = (s_1, \ldots, s_n)$ be an input sentence of n words, which is passed through a neural network that computes d-dimensional hidden representations $(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)})$, *e.g.* BERT (Devlin et al., 2019). Then, output tag weights for word i can be computed via a linear model:

$$\boldsymbol{w}^{(i)} = \boldsymbol{B}\boldsymbol{x}^{(i)} + \boldsymbol{d} \tag{1}$$

where $\boldsymbol{w}^{(i)} \in \mathbb{R}^{|T|}$ are output weights, $\boldsymbol{B} \in \mathbb{R}^{|T| \times d}$ and $\boldsymbol{d} \in \mathbb{R}^{|T|}$ are the model parameters. The prediction is simply the tag of maximum weight. In our setting, we instead use tag prototypes so that we can rely on clustering for learning. As such, the weight of a tag is proportional to the negative squared Euclidean distance with the prototype:

$$w_j^{(i)} = -\frac{1}{2}\|\boldsymbol{x}^{(i)} - \boldsymbol{c}^{(j)}\|^2 \tag{2}$$

where $\boldsymbol{c}^{(j)} \in \mathbb{R}^d$ is the *prototype* of tag $j \in [\,|T|\,]$.[2]

In practice, it can be useful to have several prototypes per tag, in particular for the O tag that gathers heterogeneous classes of words. Let k be the total number of prototypes and $\boldsymbol{C} \in \mathbb{R}^{k \times d}$ be the matrix that contains in each row a prototype of dimension d. Let $\phi : [k] \to T$ be the function that assigns to each prototype a tag and ϕ^{-1} its preimage function:

$$\phi^{-1}(t) = \{i \in [k] \mid \phi(i) = t\}\,.$$

The weight of tag $t \in T$ for word $s^{(i)}$ is defined as:

$$-\min_{j \in \phi^{-1}(t)} \frac{1}{2}\|\boldsymbol{x}^{(i)} - \boldsymbol{C}_j\|^2\,,$$

or, in other words, the weight of a given tag $t \in T$ depends on the closest prototype according to ϕ^{-1}.

Few-shot evaluation. We simulate a transfer learning scenario as described by Yang and Katiyar (2020): we first pre-train a model on a *source domain* for which there exists a large annotated dataset, and then fine-tune the model on a *target domain* using only a few labeled sentences, called the *support*. The target domain may have different mention labels than the source domain.[3] However, contrary to Yang and Katiyar (2020), we assume access to a large unlabeled dataset in the target domain, which is often easy and cheap to obtain.

For pre-training, we simply use the negative log-likelihood loss defined as:

$$\ell(\boldsymbol{w}; \boldsymbol{y}) = -\langle \boldsymbol{w}, \boldsymbol{y} \rangle + \log \sum_i \exp w_i\,,$$

[1] https://github.com/ayoubhammal/ckss4ner

[2] Models (1) and (2) are equivalent, see Appendix A.

[3] This means that the output layer used during pre-training, *i.e.* either Equation (1) or (2), is not used for the target domain.

where $\boldsymbol{y}$ is a one-hot vector indicating the gold tag. Contrary to previous works (Fritzler et al., 2019; Yang and Katiyar, 2020; Das et al., 2022), we compute tag weights using Equation (2) instead of (1), as it is more similar to the fine-tuning method.[4]

Fine-tuning data. We will denote the fine-tuning dataset as the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, where n is total number of words in the fine-tuning data (labeled and unlabeled). Moreover, the matrix $\boldsymbol{Z} \in \{0,1\}^{n \times k}$ indicates which prototypes are allowed for each word. In other words, Z_{ij} is equal to 1 if and only if prototype j is allowed for datapoint i. That is, if $i \in [n]$ is tagged with $t \in T$ (*i.e.* it is in the support), then:

$$Z_{ij} = \begin{cases} 1 & \text{if } j \in \phi^{-1}(t)\,, \\ 0 & \text{otherwise.} \end{cases}$$

Otherwise, if word $i' \in [n]$ is not tagged (*i.e.* it is not in the support), then $Z_{i'j} = 1, \forall j \in [k]$.

3 Weakly-Supervised Clustering

In our few-shot settings, we have access to an additional unlabeled dataset, that is, we learn from low-recall data where only a few mentions are annotated. In this setting, it is useful to introduce extra knowledge during training as a supervision signal. We follow Effland and Collins (2021) and impose a ratio constraint on the O tag during training.

We first recall the k-means algorithm using our notation. Then, we propose novel algorithms for the E step that allows to enforce the ratio constraint. Finally, we explain an initialization strategy that leads to fully deterministic fine-tuning method.

3.1 The k-Means Algorithm

Clustering aims to find a partition of $\boldsymbol{X}$ into k clusters that minimizes the intra-cluster dispersion:

$$\min_{\pi \in \mathcal{P}_k(\,[n]\,)} \sum_{C \in \pi} \sum_{i \in C} \|\boldsymbol{X}_i - \overline{C}\|^2\,, \qquad (3)$$

where $\overline{C} = |C|^{-1} \sum_{i \in C} \boldsymbol{X}_i$ denotes the cluster centroid. It can be shown that this is a NP-hard combinatorial problem (Dasgupta, 2008; Aloise et al., 2009). The main ideas behind the k-means algorithm are: (1) allow clusters to contain no datapoint; (2) transform the combinatorial search $\pi \in \mathcal{P}_k(\,[n]\,)$ into a continuous problem over a cluster assignation matrix $\boldsymbol{A} \in \mathbb{R}^{n \times k}$; (3) replace the dispersion around a cluster centroid by its variational formulation:

$$\sum_{i \in C} \|\boldsymbol{X}_i - \overline{C}\|^2 = \min_{\boldsymbol{c} \in \mathbb{R}^d} \sum_{i \in C} \|\boldsymbol{X}_i - \boldsymbol{c}\|^2\,. \qquad (4)$$

We obtain the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{A},\boldsymbol{C}} \; & \sum_{i \in [n]} \sum_{j \in [k]} A_{ij} \|\boldsymbol{X}_i - \boldsymbol{C}_j\|^2 + \Omega(\boldsymbol{A})\,, \\ \text{s.t.} \; & \sum_{j=1}^{k} A_{ij} = 1 \quad \forall i \in [n]\,, & (5) \\ & \boldsymbol{A} \in \mathbb{R}_+^{n \times k} \text{ and } \boldsymbol{C} \in \mathbb{R}^{k \times d}\,, & (6) \end{aligned}$$

where $\Omega(\cdot)$ is a regularizer that can be interpreted in a similar way to the regularizer in the Fenchel-Young losses framework (Blondel et al., 2020). Equation (5) ensures that each datapoint is assigned to exactly one cluster. In our setting, cluster centroids in $\boldsymbol{C}$ corresponds to (learned) prototypes. To take into account the supervision knowledge encoded in $\boldsymbol{Z}$, we add the following constraint:

$$A_{ij} \leq Z_{ij} \quad \forall i \in [n]\,, j \in [k]\,, \qquad (7)$$

i.e. if a cluster is forbidden for a datapoint, the corresponding value in $\boldsymbol{A}$ is forced to 0.

The objective of the k-means problem is non-convex (An et al., 2006, Eq. 3.1), however it is bi-convex, that is convex in $\boldsymbol{A}$ (resp. $\boldsymbol{C}$) when $\boldsymbol{C}$ is fixed (resp. $\boldsymbol{A}$). Therefore, the standard optimization method is based on Alternate Convex Search (Hastie et al., 2015, Sec. 5.9): we iteratively minimize the objective over $\boldsymbol{A}$ (E step) and over $\boldsymbol{C}$ (M step).[5] Importantly, this optimization procedure is not specific to the original k-means and applies to all problems with the same properties, *e.g.* when adding Constraint (7) and the ratio Constraint (10).

Hard k-means. If $\Omega(\boldsymbol{A}) = 0$, we obtain the standard hard k-means problem. **(E step)** Note that if we minimize over $\boldsymbol{A}$ only, the cluster distance term is constant. Let $\boldsymbol{D}$ be a matrix s.t. $D_{ij} = \|\boldsymbol{X}_i - \boldsymbol{C}_j\|^2$, then the optimal assignation is:

$$\widehat{\boldsymbol{A}}_i \in \arg\min_{\boldsymbol{e} \in \triangle(k)} \langle \boldsymbol{e}, \boldsymbol{D}_i \rangle \text{ s.t. } e_j \leq Z_{ij}, \forall j \in [k]\,, \quad (8)$$

where it is usual to choose one of optimal simplex corners in case of ties. This problem can be

[4] It led to better results in early experiments.

[5] E and M names comes from the EM algorithm, of which k-means is a special case (Hastie et al., 2009, Sec. 14.3.7).

solved in $\mathcal{O}(k)$ for single datapoint, hence the time-complexity is $\mathcal{O}(nk)$. **(M step)** Minimizing over cluster centroids $\boldsymbol{C}$ simply yields:

$$\widehat{\boldsymbol{C}}_j = \frac{\sum_i A_{ij} \boldsymbol{X}_i}{\sum_i A_{ij}} . \tag{9}$$

Time-complexity is $\mathcal{O}(nk)$.

Soft k-means. If $\Omega(\boldsymbol{A})$ is the negative Shannon entropy defined as follows:

$$\Omega(\boldsymbol{A}) = -H(\boldsymbol{A}) = \langle \boldsymbol{A}, \log \boldsymbol{A} \rangle - \langle \boldsymbol{A}, \boldsymbol{1} \rangle ,$$

the resulting algorithm is known as soft k-means.[6] The optimal solution of the E step becomes:

$$\widehat{A}_{ij} = \frac{Z_{ij} \exp D_{ij}}{\sum_{j'} Z_{ij'} \exp D_{ij'}} ,$$

see (Beck, 2017, Ex. 3.71). It can be computed in $\mathcal{O}(nk)$. The M step is left unchanged.

3.2 Weak Supervision via Ratio Constraint

Let r_{O} be the expected ratio of words tagged with O. To use this extra information, we can add the following constraint to the clustering problem:

$$\sum_{i \in [n]} \sum_{j \in \phi^{-1}(\text{O})} A_{ij} = n \times r_{\text{O}} . \tag{10}$$

Note that this constraint only applies to the assignment matrix $\boldsymbol{A}$, therefore it only impacts the E step. In the following, we propose novel algorithms to compute the E step with Constraint (10).

Hard k-means. The (unconstrained) E step can be seen as a graph problem: (1) construct a bipartite graph where one set of nodes corresponds to datapoints and the other to clusters, and there is an edge connecting a datapoint i with a cluster j with weight D_{ij} if and only if $Z_{ij} = 1$, see Figure 1 (left); (2) compute the one-to-many assignment of minimum weight, where each node representing a datapoint is assigned to exactly one cluster (*i.e.* it has exactly one incident edge in the solution).

To compute the solution in the constrained case, note that we can focus solely on whether a datapoint is assigned to one of the O clusters or not. That is, we can divide cluster nodes into two groups: nodes representing O clusters (*i.e.* elements of $\phi^{-1}(\text{O})$) and nodes representing other clusters. We can therefore *contract* each group into a single node, where we keep only the edge of minimum weight between a datapoint and nodes in the contracted group, and compute the solution of the constrained E step on this simpler graph, see Figure 1 (right). It is trivial to construct the solution for the original graph from a contracted graph solution.

To this end, we build vectors $\boldsymbol{d}^{(\text{O})} \in \mathbb{R}^n$ (resp. $\boldsymbol{d}^{(\text{OTHERS})} \in \mathbb{R}^n$) that indicates the distance between each datapoint and its closest O cluster (resp. non-O cluster), that is:

$$\begin{aligned} & d_i^{(\text{O})} && = \min_{j \in \phi^{-1}(\text{O})} D_{ij} && \text{s.t. } Z_{ij} = 1 \\ \text{and } & d_i^{(\text{OTHERS})} && = \min_{j \notin \phi^{-1}(\text{O})} D_{ij} && \text{s.t. } Z_{ij} = 1 , \end{aligned}$$

where the minimum is set to ∞ if the search space is empty.

Let $\boldsymbol{a} \in \{0,1\}^n$ be an assignation vector to the O group of clusters, *i.e.* $a_i = 1$ if and only if $\boldsymbol{x}^{(i)}$ is assigned to a O cluster. Then, there is a one-to-one mapping between solutions $\widehat{\boldsymbol{A}}$ of the constrained E step and solutions $\widehat{\boldsymbol{a}}$ of the following problem:

$$\begin{aligned} & \min_{\boldsymbol{a}} \langle \boldsymbol{a}, \boldsymbol{d}^{(\text{O})} \rangle + \langle 1 - \boldsymbol{a}, \boldsymbol{d}^{(\text{OTHERS})} \rangle \\ & \text{s.t.} \sum_{i \in [n]} a_i = n \times r_{\text{O}} \quad \text{and} \quad \boldsymbol{a} \in \{0,1\}^n . \end{aligned}$$

The objective can be rewritten as follows:

$$\begin{aligned} & \langle \boldsymbol{a}, \boldsymbol{d}^{(\text{O})} \rangle + \langle 1 - \boldsymbol{a}, \boldsymbol{d}^{(\text{OTHERS})} \rangle \\ & = \langle \boldsymbol{a}, \underbrace{\boldsymbol{d}^{(\text{O})} - \boldsymbol{d}^{(\text{OTHERS})}}_{= \boldsymbol{d}'} \rangle + \underbrace{\langle \boldsymbol{1}, \boldsymbol{d}^{(\text{OTHERS})} \rangle}_{\text{constant}} , \end{aligned}$$

where the second term is constant. Computing the optimal $\widehat{\boldsymbol{a}}$ is therefore reduced to find the $n \times r_{\text{O}}$ smallest values in the penalized distance vector $\boldsymbol{d}'$. It is then trivial to build $\widehat{\boldsymbol{A}}$ from $\widehat{\boldsymbol{a}}$ by inspecting which edges was kept in the contraction step. Time-complexity is $\mathcal{O}(nk + n \log n)$ since it requires a partial sort of $\boldsymbol{d}'$.

Soft k-means. Constraints (5), (7) and (10) can be rewritten as inclusion in the intersection of the following affine subspaces:

$$\begin{aligned} S^{(1)} &= \left\{ \boldsymbol{A} \in \mathbb{R}_+^{n \times k} \,\middle|\, \forall i,j : Z_{ij} = 0 \Leftrightarrow A_{ij} = 0 \right\} \\ S^{(2)} &= \left\{ \boldsymbol{A} \in \mathbb{R}_+^{n \times k} \,\middle|\, \boldsymbol{A}\boldsymbol{1} = \boldsymbol{1} \right\} \\ S^{(3)} &= \left\{ \boldsymbol{A} \in \mathbb{R}_+^{n \times k} \,\middle|\, \sum_{i \in [n]} \sum_{j \in \phi^{-1}(\text{O})} A_{ij} = n \times r_{\text{O}} \right\} \end{aligned}$$

i.e. $\boldsymbol{A} \in S^{(1)} \cap S^{(2)} \cap S^{(3)}$. We can rewrite the E step as a KL projection into this intersection:

$$\arg\min_{\boldsymbol{A}} \langle \boldsymbol{A}, \boldsymbol{D} \rangle - H(\boldsymbol{A}) \text{ s.t. (7), (5) and (10)}$$

[6] Not to be confused with fuzzy clustering that optimizes a different objective (Dunn, 1973; Bezdek, 1981).

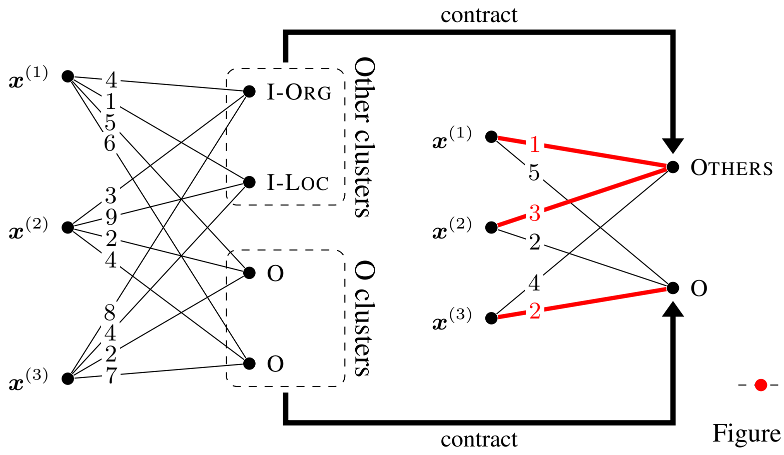

Figure 1: Illustration of the E step with ratio constraints. In the two bipartite graphs, left (resp. right) nodes represents datapoints (resp. clusters). The left graph is the full graph, where edge weights indicate distances between nodes and clusters. By contracting the two sets of clusters, we obtain a new graph, on which we can run the E step with a ratio constraint for the contracted O cluster (ratio is set to $1/3$ in the example). Thick red edges indicate the optimal solution. Note that, without the ratio constraint, $x^{(2)}$ would be assigned to the O cluster.

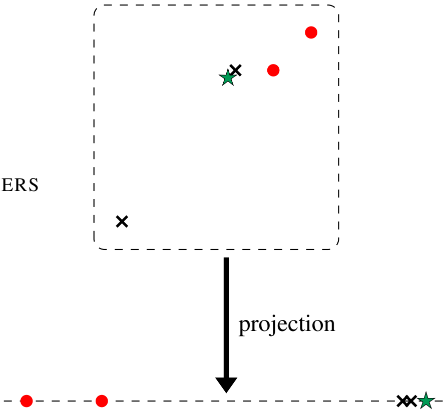

Figure 2: Illustration of the benefit of subspace selection. **(top)** Data in its original 2D space. We assume the constrained clustering results in two clusters: one containing the two black crosses and the other containing the two red circles. Let the green star be a test point. Intuitively, it should be classified in the black crosses cluster, however, it is closer to the other cluster centroid! **(bottom)** Data after projection in a 1D space. The test point is now correctly classified.

$$= \arg\min_{\boldsymbol{A}} \; KL[\boldsymbol{A} | \exp(-\boldsymbol{D})] \text{ s.t. } \boldsymbol{A} \in \bigcap_{i=1}^{3} S^{(i)}$$

This problem can be (approximatively) solved using the iterative Bregman projection algorithm (Bregman, 1967; Censor and Zenios, 1997), which have recently been popular in the optimal transport literature (Benamou et al., 2015). We iteratively project the current estimate into $S^{(1)} \cap S^{(2)}$ and $S^{(1)} \cap S^{(3)}$. More details are given in Appendix B.

3.3 Initialization

It is well known that k-means solution heavily depends on initialization (Bradley and Fayyad, 1998), and several runs with different random initializations may produce quite different results. Therefore, we opt for a deterministic approach for cluster center initialization. An important advantage of our approach is that it improves reproducibility.

We assume that for each tag $t \in T$, there exists at least $|\phi^{-1}(t)|$ words annotated with t in the dataset. If $|\phi^{-1}(t)| = 1$, we initialize the cluster centroid as the average of hidden representation of words labeled with t. Otherwise, we rely on greedy agglomerative hierarchical clustering using the Ward linkage strategy (Duda et al., 2000, Sec. 10.9). We cut the dendrogram to obtain $|\phi^{-1}(t)|$ clusters whose centroids will serve as initial centroids for the k-means procedure.

4 Subspace Selection

Although constrained clustering is convenient for weakly-supervised few-shot learning, it can lead to problems inherent to the clustering assumption: the property that each datapoint is assigned to its closest neighbor may not be satisfied in the training data due to ratio or supervision constraints. At test time, this may result in incorrect predictions. To bypass this issue, we jointly learn a transformation of the data so that clusters are well separated, see Figure 2. We focus on subspace selection, *i.e.* the transformation is restricted to a linear projection.

4.1 Problem Definition

Let $\boldsymbol{U}^\top \in \mathbb{R}^{p \times d}$, be a projection matrix of rank $p \leq d$. Then, $\boldsymbol{x}' = \boldsymbol{U}^\top \boldsymbol{x} \in \mathbb{R}^p$ is a projection of $\boldsymbol{x} \in \mathbb{R}^d$ into the p-dimensional subspace defined by the linear map.[7] The new joint constrained clustering an subspace selection problem is:

$$\min_{\boldsymbol{A},\boldsymbol{C},\boldsymbol{U}} \sum_{i \in [n]} \sum_{j \in [k]} A_{ij} \| \boldsymbol{U}^\top \boldsymbol{X}_i - \boldsymbol{U}^\top \boldsymbol{C}_j \|^2 + \Omega(\boldsymbol{A})$$
$$\text{s.t.} \quad (5), (6) \text{ and } (7).$$

One issue with this problem formulation is that it has a trivial but non interesting global optimum for

[7] Note that this is equivalent to defining a custom metric $\|\boldsymbol{U}^\top \boldsymbol{x} - \boldsymbol{U}^\top \boldsymbol{c}\|^2 = (\boldsymbol{x}-\boldsymbol{c})^\top \boldsymbol{U}\boldsymbol{U}^\top (\boldsymbol{x}-\boldsymbol{c}) = \|\boldsymbol{x}-\boldsymbol{c}\|^2_{\boldsymbol{U}\boldsymbol{U}^\top}$, called the Mahalanobis distance parameterized by $\boldsymbol{U}\boldsymbol{U}^\top$.

the first term by setting $\boldsymbol{U} = \boldsymbol{0}$, *i.e.* collapsing all points, as there is no constraint on $\boldsymbol{U}$.

Given a cluster assignment matrix $\boldsymbol{A}$, we assume $\widehat{\boldsymbol{C}}$ are the optimal centroids given by Equation (9). To simplify notation, the dependency on $\boldsymbol{A}$ of $\widehat{\boldsymbol{C}}$ is not explicitly written. The total, within-class and between-class scatter (correlation) matrices are:

$$\begin{aligned}
\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{U}} &= \sum_{i\in[n]} (\boldsymbol{U}^\top \boldsymbol{X}_i - \boldsymbol{U}^\top \overline{\boldsymbol{x}})(\boldsymbol{U}^\top \boldsymbol{X}_i - \boldsymbol{U}^\top \overline{\boldsymbol{x}})^\top \\
\boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{U},\boldsymbol{A}} &= \sum_{j\in[k]} \sum_{i\in n[n]} A_{ij} \; (\boldsymbol{U}^\top \boldsymbol{X}_i - \boldsymbol{U}^\top \widehat{\boldsymbol{C}}_j) \\
&\qquad\qquad\qquad \times (\boldsymbol{U}^\top \boldsymbol{X}_i - \boldsymbol{U}^\top \widehat{\boldsymbol{C}}_j)^\top \\
\boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{U},\boldsymbol{A}} &= \sum_{j\in[k]} \sum_{i\in n[n]} A_{ij} \; (\boldsymbol{U}^\top \widehat{\boldsymbol{C}}_j - \boldsymbol{U}^\top \overline{\boldsymbol{x}}) \\
&\qquad\qquad\qquad \times (\boldsymbol{U}^\top \widehat{\boldsymbol{C}}_j - \boldsymbol{U}^\top \overline{\boldsymbol{x}})^\top
\end{aligned}$$

where $\overline{\boldsymbol{x}} = n^{-1} \sum_{i\in[n]} \boldsymbol{X}_i$ is the sample mean. $\mathbf{tr}(\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{U}})$, $\mathbf{tr}(\boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{U},\boldsymbol{A}})$ and $\mathbf{tr}(\boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{U},\boldsymbol{A}})$ correspond to data, intra-cluster and inter-cluster dispersion.

The following equalities holds (Appendix C.1):

$$\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{U}} = \boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{U},\boldsymbol{A}} + \boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{U},\boldsymbol{A}} \tag{11}$$

$$\mathbf{tr}(\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{U}}) = \mathbf{tr}(\boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{U},\boldsymbol{A}}) + \mathbf{tr}(\boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{U},\boldsymbol{A}}) \tag{12}$$

If $\boldsymbol{U}$ *is fixed to the identity matrix* $\boldsymbol{I}$ (*i.e.* no learned subspace selection), the left-hand side is constant as it does not depend on the clustering: minimizing the intra-cluster dispersion is equivalent to maximizing the inter-cluster dispersion, so there is no cluster collapse. When jointly learning $\boldsymbol{U}$, we propose to fix the expected data dispersion as follows:[8]

$$\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{U}} = \boldsymbol{I} \quad \Leftrightarrow \quad \boldsymbol{U}^\top \boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{I}} \boldsymbol{U} = \boldsymbol{I}. \tag{13}$$

which prevent data and clusters collapse.

4.2 Optimization Algorithm

We follow an alternate convex search procedure where variables are visited in order $\boldsymbol{A} \rightarrow \boldsymbol{C} \rightarrow \boldsymbol{U}$. Minimizing over $\boldsymbol{A}$ requires to take into account for the projection when computing matrix $\boldsymbol{D}$ in Equation (8), *i.e.* we set $D_{ij} = \|\boldsymbol{U}^\top \boldsymbol{X}_i - \boldsymbol{U}^\top \boldsymbol{C}_j\|^2$. Optimization over $\boldsymbol{C}$ is left unchanged, see App. D.

We are left with optimization over $\boldsymbol{U}$. Similarly to Equation (13), we can rewrite the objective as:

$$\mathbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_{\boldsymbol{I},\boldsymbol{A}} \boldsymbol{U}) + \Omega(\boldsymbol{A})\,.$$

Ignoring the constant term, the Lagrangian is:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{U}, \boldsymbol{\Lambda}) = &\mathbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}^{(w)}_{\boldsymbol{I},\boldsymbol{A}} \boldsymbol{U}) \\
&- \mathbf{tr}(\boldsymbol{\Lambda}^\top (\boldsymbol{U}^\top \boldsymbol{S}^{(t)}_{\boldsymbol{I}} \boldsymbol{U} - \boldsymbol{I})),
\end{aligned}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{p\times p}$ are dual variables associated with Constraint (13), a.k.a. Lagrangian multipliers. $\boldsymbol{\Lambda}$ is implicitly constrained to be diagonal at optimality (Ghojogh et al., 2023, App. B). By stationarity (*i.e.* differentiating $\mathcal{L}$ w.r.t. $\boldsymbol{U}$), a primal-dual pair of variable $\widehat{\boldsymbol{U}}$ and $\widehat{\boldsymbol{\Lambda}}$ are minimizer if and only if:

$$\boldsymbol{S}^{(w)}_{\boldsymbol{I},\boldsymbol{A}} \widehat{\boldsymbol{U}} = \boldsymbol{S}^{(t)}_{\boldsymbol{I}} \widehat{\boldsymbol{U}} \widehat{\boldsymbol{\Lambda}}, \tag{14}$$

which is a generalized eigenvalue problem on pair of matrices $(\boldsymbol{S}^{(w)}_{\boldsymbol{I},\boldsymbol{A}}, \boldsymbol{S}^{(t)}_{\boldsymbol{I}})$: columns of $\widehat{\boldsymbol{U}}$ are eigenvectors, and values in the diagonal of $\widehat{\boldsymbol{\Lambda}}$ are eigenvalues (Parlett, 1998; Golub and Van Loan, 2013).

As we have a minimization problem, the optimal solution is composed of the p smallest eigenvalues. They can be computed in $\mathcal{O}((n+k)d^2 + d^3)$.

Projection dimension. We are left with one question: how to choose the projection dimension p? Note that given enough training data ($n \gg d$), which is the case in practice, the total scatter $\boldsymbol{S}^{(t)}_{\boldsymbol{I}}$ will be of full rank, that is invertible. We can rewrite Equation (14) as follows (see Appendix E):

$$\underbrace{(\boldsymbol{S}^{(t)}_{\boldsymbol{I}})^{-1} \boldsymbol{S}^{(b)}_{\boldsymbol{I},\boldsymbol{A}}}_{=\boldsymbol{S}} \widehat{\boldsymbol{U}} = \widehat{\boldsymbol{U}}\, \underbrace{(\boldsymbol{I} - \widehat{\boldsymbol{\Lambda}})}_{=\widehat{\boldsymbol{\Lambda}}'} \tag{15}$$

which is equivalent to computing eigenvalues $\widehat{\boldsymbol{\Lambda}}'$ of matrix $\boldsymbol{S}$. We set p to the maximum number of non-null eigenvalues we can get, that is:

$$\begin{aligned}
p = \operatorname{rank}(\boldsymbol{S}) &= \operatorname{rank}\left((\boldsymbol{S}^{(t)}_{\boldsymbol{I}})^{-1} \boldsymbol{S}^{(b)}_{\boldsymbol{I},\boldsymbol{A}} \right) \\
&= \min\left(\operatorname{rank}\left((\boldsymbol{S}^{(t)}_{\boldsymbol{I}})^{-1} \right), \operatorname{rank}\left(\boldsymbol{S}^{(b)}_{\boldsymbol{I},\boldsymbol{A}} \right) \right) \\
&= \min(d, k-1) = k - 1\,.
\end{aligned}$$

As the rank of scatter matrix $\boldsymbol{S}^{(b)}_{\boldsymbol{I},\boldsymbol{A}}$ is equal to $k-1$ in non degenerated cases.[9]

5 Related Work

Few-shot learning. A common approach for few-shot learning is to learn a neural network based metric distance (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018, *inter alia*). Although our approach can also be interpreted as metric learning, we simplify the process by restricting ourselves to using the euclidean distance after projection in a subspace, where the subspace projection is learned.

In the case of NER, Fritzler et al. (2019) adapted the prototypical network of Snell et al. (2017). Yang and Katiyar (2020) rely on nearest-neighbor

[8] See Appendix C.2 for proof of the equivalence.

[9] It is a sum of k rank-1 matrices that are tied by the mean.

Model	**1-shot**				**5-shot**			
	A	**B**	**C**	**Avg.**	**A**	**B**	**C**	**Avg.**
Proto †	19.3±3.9	22.7±8.9	18.9±7.9	20.3	30.5±3.5	38.7±5.6	41.1±3.3	36.7
NNShot †	28.5±9.2	27.3±12.3	21.4±9.7	25.7	44.0±2.1	51.6±5.9	47.6±2.8	47.7
StructShot †	30.5±12.3	28.8±11.2	20.8±9.9	26.7	47.5±4.0	53.0±7.9	48.7±2.7	49.8
CONTaiNER †	32.2±5.3	30.9±11.6	32.9±12.7	32.0	51.2±5.9	55.9±6.2	61.5±2.7	56.2
+ Viterbi †	32.4±5.1	30.9±11.6	33.0±12.8	32.1	51.2±6.0	56.0±6.2	61.5±2.7	56.2
Our reproduction on our support sets								
NNShot	23.9±10.0	28.2±8.1	23.0±8.5	25.0	37.9±6.1	50.6±6.6	38.8±3.5	42.4
StructShot	24.6±10.2	28.2±8.0	23.4±8.6	25.4	40.2±6.0	50.9±6.8	41.5±4.1	44.2
K-Means with subspace selection using unlabeled dev + train sets								
Hard clustering (# O-clusters = 10, # I-clusters = 1)								
$r_O = NA$	39.5±11.6	60.3±7.8	46.6±10.5	48.8	36.4±10.3	70.1±4.4	57.6±6.2	54.7
$r_O = 0.95, 0.96, 0.93$	**43.5±12.8**	**60.6**±6.4	45.1±11.3	**49.7**	**54.5±13.8**	69.2±7.8	60.1±6.3	**61.3**
Soft clustering (# O-clusters = 10, # I-clusters = 1)								
$r_O = NA$	39.4±11.6	60.3±7.8	46.5±10.6	48.7	35.9±10.5	70.0±4.4	57.6±6.2	54.5
$r_O = 0.95, 0.96, 0.93$	40.1±11.7	57.7±13.5	**47.1±11.5**	48.3	47.1±10.7	**72.3**±5.4	**63.5±5.9**	61.0

Table 1: Results for the tag set extension experiments reported in F1-score. Results marked with † are taken from Das et al. (2022), and are evaluated on different support sets than ours. For our evaluation, we generated 10 support sets following the sampling algorithm proposed by Yang and Katiyar (2020). Ratio constraints are noted in order of datasets, i.e. $r_O = 0.95, 0.96, 0.93$ means that models evaluated on tag set A, tag set B, tag set C use a 0.95, 0.96, 0.93 ratio respectively.

classification along with a meta-transition matrix learned from the source task but with simpler IO transitions. Unfortunately, the later requires to tune a temperature hyper-parameter on the target domain, which is not realistically possible in the few-shot setting where there is no development set. Das et al. (2022) extended this approach by fine-tuning on both of the source dataset and target support using a contrastive loss function. Chen et al. (2023) proposed to augment the input with output label information, similarly to the GLiNER approach (Zaratiana et al., 2024). Closer to our work, Hou et al. (2020) rely on a target task specific linear projection as proposed by Yoon et al. (2019), but they cannot benefit from extra unlabeled data.

Ratio constraints. Using a weak supervision signal in the E step has been known as posterior regularization in the case of generative models (Ganchev et al., 2010). However, generic application of this framework rely on costly gradient descent to compute the solution of the E step, whereas we propose a polynomial analytical solution for our hard k-means case and an efficient approach based on iterative Bregman projections for the soft case,. Previous work on ratio constraints for k-means reduced the E step to transportation problems (Ng, 2000; Bradley et al., 2000) but rely on generic algorithms, whereas we propose an efficient algorithm that benefit from the structure of our ratio constraint.

Subspace selection. Joint k-means and subspace selection is known as *discriminative k-means* (Ding and Li, 2007; Ye et al., 2007a; De la Torre and Kanade, 2006; Ye et al., 2007b). We depart from previous work (*e.g.* Ding and Li, 2007) by proposing an grounded and well-defined approaches instead of the mere combination of independent steps.

6 Experiments

We follow previous work (Yang and Katiyar, 2020; Das et al., 2022) and use OntoNotes5 (Weischedel et al., 2013) as generic data (news, conversational telephone speech, weblogs, usenet newsgroups, broadcast, talk shows) together with CoNLL2003 (Sang and De Meulder, 2003) and WNUT17 (Derczynski et al., 2017) as specialized data (news and social media, respectively).[10]

We initialise the model with `base-bert-cased` (Devlin et al., 2019). We pre-train on the source domain for 3 epochs using a 5×10^{-5} learning rate with a linear decay. For consistency, the pre-training is performed with the IO tagging scheme. If a word is splitted into subtokens, we average the hidden representation of first and last subtokens.

[10] Previous works also evaluate additionally on I2B2 2014 (Stubbs and Özlem Uzuner, 2015) for the medical domain. Despite sharing the support sets they used on this dataset, they do not share the preprocessing steps they employed for sentence segmentation and tokenization, which hinder our comparative evaluation.

Model	1-shot			5-shot		
	CoNLL	WNUT17	Avg.	CoNLL	WNUT17	Avg.
Proto †	49.9±8.6	17.4±4.9	33.7	61.3±9.1	22.8±4.5	42.1
NNShot †	61.2±10.4	22.7±7.4	41.9	74.1±2.3	27.3±5.4	50.7
StructShot †	62.4±10.5	24.2±8.0	43.3	74.8±2.4	30.4±6.5	52.6
CONTaiNER †	57.8±10.7	24.2±2.9	41.0	72.8±2.0	27.7±2.2	50.3
+ Viterbi †	61.2±10.7	27.5±1.9	44.4	**75.8±2.7**	32.5±3.8	54.2
Hard clustering (# O-clusters = 10, # I-clusters = 1)						
$r_O = NA$	65.4±12.0	20.9±8.2	43.2	75.7±2.1	26.5±5.3	51.1
$r_O = 0.80, 0.90$	62.6±10.6	23.6±7.1	43.1	71.9±3.0	29.3±3.1	50.6
$r_O = 0.85, 0.95$	**66.4±13.1**	**28.9±9.4**	**47.7**	**75.8±2.6**	**39.0±3.6**	**57.4**
Soft clustering (# O-clusters = 10, # I-clusters = 1)						
$r_O = NA$	65.4±12.0	20.8±8.1	43.1	75.7±2.1	26.5±5.3	51.1
$r_O = 0.80, 0.90$	65.6±12.5	26.6±8.7	46.1	75.5±2.5	35.7±3.5	55.6
$r_O = 0.85, 0.95$	65.6±12.5	26.8±8.8	46.2	75.4±2.5	35.8±3.5	55.6

Table 2: Results for the domain adaptation experiments reported in F1-score. Results marked with † are taken from Das et al. (2022) and evaluation of our proposed method is done on the same 10 supports. Ratio constraints are noted in the order of datasets, i.e. $r_O = 0.80, 0.90$ means that the first dataset, CoNLL, is tested with a ratio of 0.80 and the second dataset, WNUT17, is tested with a ratio of 0.90.

When evaluating in the few-shot settings, we use the train set and development set belonging to each support as unlabeled data. For k-means, we use 10 iterations of the alternate convex search procedure. Note that this step takes approximately 6 minutes on CPU when using the full CoNLL dataset. We have one cluster per entity type (I clusters), and we fix the number of clusters for the O tag to 10. As we cannot assume to know the true ratio of O tags, we evaluate our approach with both under- and over-estimations.

Prediction. Given word s with hidden representation $\boldsymbol{x}$, we simply predict the tag associated with the closest cluster:[11]

$$\hat{y}(\boldsymbol{x}) \in \phi\left(\operatorname*{arg\,min}_{j\in[k]} \|\boldsymbol{U}^\top \boldsymbol{x} - \boldsymbol{U}^\top \boldsymbol{C}_j\|^2\right).$$

6.1 Few-Shot Settings

Tag set extension. This setting evaluates the performance of the model on a set of new tags, but without changing the input data domain. For a group $T' \subset T \setminus \{\text{O}\}$ of tags, we (1) pre-train the model using only mentions of type $T \setminus T'$ in the training data and (2) evaluate in a few-shot setting on mention of types T'. We follow Yang and Katiyar (2020) and use Ontonotes5 three different sets T', reported in Appendix F. We sample 10 support sets for each T' using the algorithm provided by Yang and Katiyar (2020). We compare to the results of previous works evaluated on their own support sets since those are not publicly available.

Domain transfer. This setting evaluates the performance of the model on a new set of tags semantically different than those seen during pre-training, and on a different input data source. To this end, we use Ontonotes5 as a for pre-training, and evaluate few-shot performances on CoNLL2003 and WNUT17. We use the same support sets as Das et al. (2022).

6.2 Results

Tables 1 and 2 summarize the results for the tag set extension and the domain transfer experiments, respectively. The constrained k-means with subspace selection algorithm performs considerably better than other baseline approaches across all experimental settings, with the biggest improvements observed on the tag set extension setting.

Hard and soft assignments result in very similar performances on the unconstrained version of the algorithm. It is interesting then to notice that hard assignments version benefits more from the ratio information than the soft assignment one. In the contrary, the soft assignment version of the algorithm is less sensitive to the ratio constraint and seem to keep a stable performance even with suboptimal ratio. We hypothesis that this is due to the fact that without ratio constraints, the hard k-means may stick to incorrect early decisions, whereas soft-assignations allows to escape them.

Ablation results are given in Appendix G.

[11] Yang and Katiyar (2020) rely on Viterbi decoding with a transition matrix learned on pre-training data. However, it has a temperature parameter which can only be tuned on the test data. Therefore we did not adopt this decoding strategy.

7 Conclusion

We propose a novel weakly-supervised algorithm for the few-shot NER. We evaluate our approach on different scenarios and achieve state-of-the-art results. Future work could consider applying our approach to learning from partial labels (Jin and Ghahramani, 2002) and to transductive learning (Colombo et al., 2023).

Limitations

In practice, it is important to be able to differentiate between succeeding mentions of the same type using B tags. Unfortunately, including B tags is non-trivial in our approach, and solutions should be considered in future research. The same limitation happens for inner mentions in the case of nested NER. Although this is an important limitation of our approach, it is also a limitation of previous work for few-shot NER.

We were unable to compare our approach on the I2B2 dataset as authors did not release unlabeled data using their pre-processing method, nor their pre-processing scripts.

Acknowledgments

This work was done while first and last authors were researchers at ISIR in the MLIA team, both funded by the Sorbonne Center for Artificial Intelligence (SCAI). This work has received funding from the French "Agence Nationale pour la Recherche" under grant agreement InExtenso ANR-23-IAS1-0004.

References

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. 2009. Np-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248.

Le Thi Hoai An, M. Tayeb Belghiti, and Pham Dinh Tao. 2006. A new efficient algorithm based on dc programming and dca for clustering. *Journal of Global Optimization*, 37(4):593–608.

Amir Beck. 2017. *First-order methods in optimization*. SIAM.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. 2015. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138.

James C. Bezdek. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, USA.

Mathieu Blondel, André F.T. Martins, and Vlad Niculae. 2020. Learning with fenchel-young losses. *Journal of Machine Learning Research*, 21(35):1–69.

Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186, Heidelberg. Physica-Verlag HD.

Paul S Bradley, Kristin P Bennett, and Ayhan Demiriz. 2000. Constrained k-means clustering. *Microsoft Research, Redmond*, 20(0):0.

Paul S Bradley and Usama M Fayyad. 1998. Refining initial points for k-means clustering. In *ICML*, volume 98, pages 91–99. Citeseer.

Lev M Bregman. 1967. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217.

Yair Censor and Stavros Andrea Zenios. 1997. *Parallel optimization: Theory, algorithms, and applications*. Oxford University Press, USA.

Yanru Chen, Yanan Zheng, and Zhilin Yang. 2023. Prompt-based metric learning for few-shot NER. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7199–7212, Toronto, Canada. Association for Computational Linguistics.

N. Chinchor and P. Robinson. 1998. Appendix E: MUC-7 named entity task definition (version 3.5). In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

Pierre Colombo, Victor Pellegrain, Malik Boudiaf, Myriam Tami, Victor Storchan, Ismail Ayed, and Pablo Piantanida. 2023. Transductive learning for textual few-shot classification in API-based embedding models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4214–4231, Singapore. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland. Association for Computational Linguistics.

Sanjoy Dasgupta. 2008. The hardness of k-means clustering. Technical report, Department of Computer Science and Engineering, University of California, San Diego.

Fernando De la Torre and Takeo Kanade. 2006. Discriminative cluster analysis. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 241–248, New York, NY, USA. Association for Computing Machinery.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Ding and Tao Li. 2007. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 521–528, New York, NY, USA. Association for Computing Machinery.

Richard O Duda, Peter E Hart, and David G Stork. 2000. *Pattern Classification*, 2 edition. A Wiley-Interscience publication. John Wiley & Sons, Nashville, TN.

J. C. Dunn. 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

Thomas Effland and Michael Collins. 2021. Partially supervised named entity recognition via the expected entity ratio loss. *Transactions of the Association for Computational Linguistics*, 9:1320–1335.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 993–1000, New York, NY, USA. Association for Computing Machinery.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(67):2001–2049.

Benyamin Ghojogh, Fakhri Karray, and Mark Crowley. 2023. Eigenvalue and generalized eigenvalue problems: Tutorial. *Preprint*, arXiv:1903.11240.

Gene H Golub and Charles F Van Loan. 2013. *Matrix computations*. JHU press.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer New York.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. 2015. Statistical learning with sparsity. *Monographs on statistics and applied probability*, 143(143):8.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1381–1393, Online. Association for Computational Linguistics.

Rong Jin and Zoubin Ghahramani. 2002. Learning with multiple labels. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, volume 1, page 3.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 188–191.

Michael K Ng. 2000. A note on constrained k-means algorithms. *Pattern Recognition*, 33(3):515–519.

Beresford N. Parlett. 1998. *The Symmetric Eigenvalue Problem*. Society for Industrial and Applied Mathematics.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Sunita Sarawagi and William W Cohen. 2004. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *Journal of Biomedical Informatics*, 58:S20–S29. Supplement: Proceedings of the 2014 i2b2/UTHealth Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Jieping Ye, Zheng Zhao, and Huan Liu. 2007a. Adaptive distance metric learning for clustering. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7.

Jieping Ye, Zheng Zhao, and Mingrui Wu. 2007b. Discriminative k-means for clustering. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. 2019. TapNet: Neural network augmented with task-adaptive projection for few-shot learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7115–7123. PMLR.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.

A Distance and Projection Weights Equivalence

The negative squared Euclidean distance with the prototype can be re-written as:

$$\begin{aligned} w_j &= -\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{C}_j\|^2 \\ &= -\frac{1}{2}(\|\boldsymbol{x}\|^2 + \|\boldsymbol{C}_j\|^2 - 2\langle \boldsymbol{x}, \boldsymbol{C}_j\rangle) \\ &= \langle \boldsymbol{x}, \boldsymbol{C}_j\rangle - \frac{1}{2}\|\boldsymbol{C}_j\|^2 - \frac{1}{2}\|\boldsymbol{x}\|^2 \end{aligned}$$

where we can define:

$$\boldsymbol{d}' = \begin{bmatrix} -\frac{1}{2}\|\boldsymbol{C}_1\|^2 \\ \vdots \\ -\frac{1}{2}\|\boldsymbol{C}_{|T|}\|^2 \end{bmatrix},$$

and:

$$c = -\frac{1}{2}\|\boldsymbol{x}\|^2 \in \mathbb{R},$$

therefore:

$$\boldsymbol{w} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{d}' + c\,.$$

and by the constant invariance of the Softmax operation (Blondel et al., 2020, Proposition 1), we can show that:

$$\text{softmax}(\boldsymbol{C}\boldsymbol{x} + \boldsymbol{d}' + c) = \text{softmax}(\boldsymbol{C}\boldsymbol{x} + \boldsymbol{d}')$$

where $\boldsymbol{C}\boldsymbol{x} + \boldsymbol{d}'$ is a linear model.

B Soft k-Means and Ratio Constraints

In this section, we explain how to compute the E step of soft k-means with the ratio constraint. The method is based on iterative Bregman projections. We report the reader to (Benamou et al., 2015) and (Censor and Zenios, 1997) for an in-depth explanation of this method.

Definition 1 ***(Bregram divergence).*** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ be a strictly convex and continuously differentiable function. The f-Bregman divergence $D_f : \text{dom} f \times \text{int}(\text{dom} f) \to \mathbb{R}$ is defined as:*

$$D_f(\boldsymbol{p}, \boldsymbol{q}) = f(\boldsymbol{p}) - f(\boldsymbol{q}) - \langle \nabla f(\boldsymbol{q}), \boldsymbol{p} - \boldsymbol{q}\rangle.$$

Definition 2 ***(Bregman projection).*** *Let S be a set and f a strictly convex and continuously differentiable function. The Bregman projection $\text{Proj}_{f,S}$ is defined as:*

$$\text{Proj}_{f,S}(\boldsymbol{q}) \in \arg\min_{\boldsymbol{p} \in S} D_f(\boldsymbol{p}, \boldsymbol{q}).$$

Definition 3 *(Iterative Bregman projections). Let D_f be a Bregman divergence and $S = \cap_{i=1}^{k} S^{(i)}$ be a set defined as the intersection of k affine sets $S^{(i)}$. We consider problems of the following form:*

$$\mathrm{Proj}_{f,S}(\boldsymbol{q}) = \arg\min_{\boldsymbol{p}\in S} D_f(\boldsymbol{p}, \boldsymbol{q}),$$

where $\boldsymbol{q} \in \mathrm{dom} f$ is a given input. The iterative Bregman projection algorithm computes a solution of this problem as follows:

- $\boldsymbol{p}^{(0)} = \boldsymbol{q}$,
- $\forall t > 0 : \boldsymbol{p}^{(t)} = \mathrm{Proj}_{f,S^{(t)}}(\boldsymbol{p}^{(t-1)})$, *were we extend the indexing of the sets by k-periodicity, i.e. $S^{(t+k)} = S^{(t)}$.*

We have $\boldsymbol{p}^{(t)} \to \mathrm{Proj}_{f,S}(\boldsymbol{q})$ as $t \to \infty$.

Now, the constrained soft k-means problem over $\boldsymbol{A}$ is defined as:

$$\arg\min_{\boldsymbol{A}} \; KL[\boldsymbol{A} | \exp(-\boldsymbol{D})] \text{ s.t. } \boldsymbol{A} \in \bigcap_{i=1}^{3} S^{(i)}$$

This previous problem over $\boldsymbol{A}$ can be rewritten as a Bregman projection such that:

$$\arg\min_{\boldsymbol{A}\in S} KL[\boldsymbol{A} | \exp(-\boldsymbol{D})] = \mathrm{Proj}_{-H,S}(\exp(-\boldsymbol{D}))$$

such that $S = \bigcap_{i=1}^{3} S^{(i)}$. Although the full problem does not have an analytic solution, it can be solved approximately given enough iterations using iterative Bregman projections. We define two affine sets: $S^{(1)} \cap S^{(2)}$ and $S^{(1)} \cap S^{(3)}$.

The solutions of the projection over each of those sets are:

$$\begin{aligned} &\widehat{\boldsymbol{A}} \in \mathrm{Proj}_{-H,S^{(1)}\cap S^{(2)}}(\boldsymbol{A}) \\ &\iff \widehat{A}_{ij} = \frac{Z_{ij} \exp\log A_{ij}}{\sum_{j'\in[k]} Z_{ij'} \exp\log A_{ij'}}, \\ &\quad \forall i \in [n], j \in [k], \end{aligned}$$

and

$$\begin{aligned} &\widehat{\boldsymbol{A}} \in \mathrm{Proj}_{-H,S^{(1)}\cap S^{(3)}}(\boldsymbol{A}) \\ &\iff \widehat{A}_{ij} = \\ &\begin{cases} \frac{Z_{ij} n\times r_{\mathrm{O}} \exp\log A_{ij}}{\sum_{i'\in[n],j'\in\sigma^{-1}(\mathrm{O})} Z_{i'j'} \exp\log A_{i'j'}} & \text{if } j \in \sigma^{-1}(\mathrm{O}), \\ Z_{ij} \exp\log A_{ij} & \text{otherwise,} \end{cases} \\ &\quad \forall i \in [n], j \in [k]. \end{aligned}$$

From this, we can derive an iterative algorithm to solve the constrained soft k-means problem over $\boldsymbol{A}$ using iterative Bregman projections.

C Scatter Matrices

Given a cluster assignment matrix $\boldsymbol{A}$, we assume $\widehat{\boldsymbol{C}}$ are the optimal centroids given by Equation (9). To simplify notation, the dependency on $\boldsymbol{A}$ of $\widehat{\boldsymbol{C}}$ is not explicitly written. The total, within-class and between-class scatter matrices are defined as follows:

$$\begin{aligned} \boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{I},\boldsymbol{A}} &= \sum_{i\in[n]} (\boldsymbol{X}_i - \overline{\boldsymbol{x}})(\boldsymbol{X}_i - \overline{\boldsymbol{x}})^\top \\ \boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{I},\boldsymbol{A}} &= \sum_{j\in[k]} \sum_{i\in n[n]} A_{ij} (\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j)(\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j)^\top \\ \boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{I},\boldsymbol{A}} &= \sum_{j\in[k]} \sum_{i\in n[n]} A_{ij} (\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}})(\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}})^\top \end{aligned}$$

where $\overline{\boldsymbol{x}} = n^{-1} \sum_{i\in[n]} \boldsymbol{X}_i$ is the sample mean. The $\boldsymbol{I}$ in the denominator indicates we consider data in the original space. Note that $\mathbf{tr}(\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{I},\boldsymbol{A}})$, $\mathbf{tr}(\boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{I},\boldsymbol{A}})$ and $\mathbf{tr}(\boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{I},\boldsymbol{A}})$ corresponds to data, intra-cluster and inter-cluster dispersion.

C.1 Equality

In this section, we prove the following equality:

$$\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{I},\boldsymbol{A}} = \boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{I},\boldsymbol{A}} + \boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{I},\boldsymbol{A}}.$$

Although this equality in well-known in the hard cluster assignment case (e.g., Hastie et al., 2009, Sec. 14.3.5), our proof also applies to soft assignments. An important implication of this equality is that we have:

$$\mathbf{tr}(\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{I},\boldsymbol{A}}) = \mathbf{tr}(\boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{I},\boldsymbol{A}}) + \mathbf{tr}(\boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{I},\boldsymbol{A}}),$$

meaning that minimizing the intra-cluster dispersion (i.e. the k-means objective) is equivalent to maximizing the inter-cluster dispersion (i.e. finding well-separated clusters) as the data dispersion is constant.

First, note that as each row of a valid assignment matrix must sum to 1, we can write:

$$\begin{aligned} \boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{I},\boldsymbol{A}} &= \sum_{i\in[n]} (\boldsymbol{X}_i - \overline{\boldsymbol{x}})(\boldsymbol{X}_i - \overline{\boldsymbol{x}})^\top \\ &= \sum_{i\in[n]} \underbrace{\left(\sum_{j\in[k]} A_{ij} \right)}_{=1} (\boldsymbol{X}_i - \overline{\boldsymbol{x}})(\boldsymbol{X}_i - \overline{\boldsymbol{x}})^\top \\ &= \sum_{j\in[k]} \sum_{i\in[n]} A_{ij} (\boldsymbol{X}_i - \overline{\boldsymbol{x}})(\boldsymbol{X}_i - \overline{\boldsymbol{x}})^\top. \end{aligned}$$

We now substract and add $\widehat{\boldsymbol{C}_j}$ inside the two terms of the matrix multiplication, and then expand the multiplication:

$$
\begin{aligned}
&= \sum_{j\in[k]}\sum_{i\in[n]} A_{ij}\ \ (\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j + \widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}}) \\
&\qquad\qquad\qquad \times(\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j + \widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}})^\top \\
&= \underbrace{\sum_{j\in[k]}\sum_{i\in[n]} A_{ij}(\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j)(\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j)^\top}_{=\boldsymbol{S}^{(\mathrm{w})}} \\
&\quad + \underbrace{\sum_{j\in[k]}\sum_{i\in[n]} A_{ij}(\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}})(\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}})^\top}_{=\boldsymbol{S}^{(\mathrm{b})}} \\
&\quad + \sum_{j\in[k]}\sum_{i\in[n]} A_{ij}(\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j)(\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}})^\top \\
&\quad + \sum_{j\in[k]}\sum_{i\in[n]} A_{ij}(\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}})(\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j)^\top
\end{aligned}
$$

We are left with showing that the two last terms are null.

We show that the fourth term is null, the third one follows a similar derivation. We can move out the term that doesn't depends on i from the sum, and then expand the factorization by A_{ij}:

$$
\begin{aligned}
&\sum_{j\in[k]}\sum_{i\in[n]} A_{ij}(\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}})(\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j)^\top \\
&= \sum_{j\in[k]}(\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}}) \sum_{i\in[n]} A_{ij}(\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j)^\top \\
&= \sum_{j\in[k]}(\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}}) \left(\sum_{i\in[n]} A_{ij}\boldsymbol{X}_i - \sum_{i\in[n]} A_{ij}\widehat{\boldsymbol{C}}_j\right)^\top \\
&= \sum_{j\in[k]}(\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}}) \left(\sum_{i\in[n]} A_{ij}\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j\sum_{i\in[n]} A_{ij}\right)^\top .
\end{aligned}
$$

If we replace the leftmost occurrence of $\widehat{\boldsymbol{C}}_j$ using Equation 9, we can then see that the second term of the matrix multiplication is equal to the null vector:

$$
\begin{aligned}
&= \sum_{j\in[k]}(\widehat{\boldsymbol{C}}_j - \overline{\boldsymbol{x}}) \\
&\quad \times \underbrace{\left(\sum_{i\in[n]} A_{ij}\boldsymbol{X}_i - \underbrace{\frac{\sum_{i'\in n} A_{i'j}\boldsymbol{X}_i}{\sum_{i'\in n} A_{i'j}}}_{=\widehat{\boldsymbol{C}}_j}\sum_{i\in[n]} A_{ij}\right)^\top}_{=\boldsymbol{0}} ,
\end{aligned}
$$

which ends the proof.

C.2 Scatter Matrices and Subspace Projection

We now turns to the case were the data is projected into a subspace using matrix $\boldsymbol{U}^\top$. The total scatter matrix in this case can be written as follows:

$$
\begin{aligned}
\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{U}} &= \sum_{i\in[n]}(\boldsymbol{U}^\top\boldsymbol{X}_i - \boldsymbol{U}^\top\overline{\boldsymbol{x}})(\boldsymbol{U}^\top\boldsymbol{X}_i - \boldsymbol{U}^\top\overline{\boldsymbol{x}})^\top \\
&= \sum_{i\in[n]}\boldsymbol{U}^\top(\boldsymbol{U}^\top\boldsymbol{X}_i - \boldsymbol{U}^\top\overline{\boldsymbol{x}})(\boldsymbol{X}_i - \overline{\boldsymbol{x}})^\top\boldsymbol{U} \\
&= \boldsymbol{U}^\top\left(\sum_{i\in[n]}(\boldsymbol{U}^\top\boldsymbol{X}_i - \boldsymbol{U}^\top\overline{\boldsymbol{x}})(\boldsymbol{X}_i - \overline{\boldsymbol{x}})^\top\right)\boldsymbol{U} \\
&= \boldsymbol{U}^\top\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{I},\boldsymbol{A}}\boldsymbol{U} .
\end{aligned}
$$

Similarly, we have:

$$
\begin{aligned}
&\boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{U}} = \boldsymbol{U}^\top\boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{I},\boldsymbol{A}}\boldsymbol{U} \\
\text{and}\quad &\boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{U}} = \boldsymbol{U}^\top\boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{I},\boldsymbol{A}}\boldsymbol{U} ,
\end{aligned}
$$

and therefore the following relation trivially holds in the projected case too:

$$
\boldsymbol{S}^{(\mathrm{t})}_{\boldsymbol{U},\boldsymbol{A}} = \boldsymbol{S}^{(\mathrm{w})}_{\boldsymbol{U},\boldsymbol{A}} + \boldsymbol{S}^{(\mathrm{b})}_{\boldsymbol{U},\boldsymbol{A}} .
$$

D M Step with Joint Subpsace Selection

The solution of the M step is left unchanged when the distance depends on a projection matrix $\boldsymbol{U}$. By first order optimality conditions, $\widehat{\boldsymbol{C}}_j$ is a minimizer if and only if:

$$
\begin{aligned}
\nabla_{\widehat{\boldsymbol{C}}_j}\sum_{\substack{i\in[n],\\ j\in[k]}} A_{ij}\|\boldsymbol{U}^\top\boldsymbol{X}_i - \boldsymbol{U}^\top\widehat{\boldsymbol{C}}_j\|^2] &= \boldsymbol{0} \\
2\sum_{i\in[n]} A_{ij}\boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{X}_i - \boldsymbol{U}^\top\widehat{\boldsymbol{C}}_j) &= \boldsymbol{0} \\
2\boldsymbol{U}\boldsymbol{U}^\top\sum_{i\in[n]} A_{ij}(\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j) &= \boldsymbol{0} \\
\sum_{i\in[n]} A_{ij}(\boldsymbol{X}_i - \widehat{\boldsymbol{C}}_j) &= \boldsymbol{0} \\
\widehat{\boldsymbol{C}}_j &= \frac{\sum_{i\in[n]} A_{ij}\boldsymbol{X}_i}{\sum_{i\in[n]} A_{ij}} .
\end{aligned}
$$

E Subspace Dimension

We derive Equation (15) as follows. Remember that stationarity condition, Equation (14), is:

$$
\boldsymbol{S}^{(w)}_{\boldsymbol{I},\boldsymbol{A}}\widehat{\boldsymbol{U}} = \boldsymbol{S}^{(t)}_{\boldsymbol{I}}\widehat{\boldsymbol{U}}\widehat{\boldsymbol{\Lambda}}
$$

By assumption, $\boldsymbol{S}_{\boldsymbol{I}}^{(t)}$ is of full rank, therefore invertible. We multiplty both sides by its inverse, and rewrite $\boldsymbol{S}_{\boldsymbol{I},\boldsymbol{A}}^{(w)}$ using Equality (11):

$$(\boldsymbol{S}_{\boldsymbol{I}}^{(t)})^{(-1)}(\boldsymbol{S}_{\boldsymbol{I}}^{(t)} - \boldsymbol{S}_{\boldsymbol{I},\boldsymbol{A}}^{(b)})\widehat{\boldsymbol{U}} = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{\Lambda}}$$

By expanding and re-arranging terms, we obtain:

$$\begin{aligned}\widehat{\boldsymbol{U}} - (\boldsymbol{S}_{\boldsymbol{I}}^{(t)})^{(-1)}\boldsymbol{S}_{\boldsymbol{I},\boldsymbol{A}}^{(b)}\widehat{\boldsymbol{U}} &= \widehat{\boldsymbol{U}}\widehat{\boldsymbol{\Lambda}}\\ (\boldsymbol{S}_{\boldsymbol{I}}^{(t)})^{(-1)}\boldsymbol{S}_{\boldsymbol{I},\boldsymbol{A}}^{(b)}\widehat{\boldsymbol{U}} &= \widehat{\boldsymbol{U}} - \widehat{\boldsymbol{U}}\widehat{\boldsymbol{\Lambda}}\\ \underbrace{(\boldsymbol{S}_{\boldsymbol{I}}^{(t)})^{(-1)}\boldsymbol{S}_{\boldsymbol{I},\boldsymbol{A}}^{(b)}}_{=\boldsymbol{S}}\widehat{\boldsymbol{U}} &= \widehat{\boldsymbol{U}}\underbrace{(\boldsymbol{I} - \widehat{\boldsymbol{\Lambda}})}_{=\widehat{\boldsymbol{\Lambda}}'}\end{aligned}$$

which is equivalent to computing eigenvalues $\widehat{\boldsymbol{\Lambda}}'$ of matrix $\boldsymbol{S}$.

F Tag Set Extension Splits

The list of type T' used for few-shot adaptation are:

Group A {ORG, NORP, ORDINAL, WORK OF ART, QUANTITY, LAW}

Group B {GPE, CARDINAL, PERCENT, TIME, EVENT, LANGUAGE}

Group C {PERSON, DATE, MONEY, LOC, FAC, PRODUCT}

G Ablation Results

Results using different unlabeled datasets are given in Table 3. We use either the full train and dev data as unlabeled data, or only the dev data, or no unlabeled data at all (i.e. the E step becomes trivial, and the algorithm reduces to subspace selection).

Results without subspace selection are given in Table 4. We observe that subspace selection improves results.

Model	**1-shot**			**5-shot**		
	CoNLL	**WNUT17**	**Avg.**	**CoNLL**	**WNUT17**	**Avg.**
Proto †	49.9±8.6	17.4±4.9	33.7	61.3±9.1	22.8±4.5	42.1
NNShot †	61.2±10.4	22.7±7.4	41.9	74.1±2.3	27.3±5.4	50.7
StructShot †	62.4±10.5	24.2±8.0	43.3	74.8±2.4	30.4±6.5	52.6
CONTaiNER †	57.8±10.7	24.2±2.9	41.0	72.8±2.0	27.7±2.2	50.3
+ Viterbi †	61.2±10.7	27.5±1.9	44.4	**75.8**±**2.7**	32.5±3.8	54.2
K-Means with subspace selection using unlabeled dev + train sets						
Hard clustering (# O-clusters = 10, # I-clusters = 1)						
$r_O = NA$	65.4±12.0	20.9±8.2	43.2	75.7±2.1	26.5±5.3	51.1
$r_O = 0.80, 0.90$	62.6±10.6	23.6±7.1	43.1	71.9±3.0	29.3±3.1	50.6
$r_O = 0.85, 0.95$	**66.4**±**13.1**	**28.9**±**9.4**	**47.7**	**75.8**±**2.6**	**39.0**±**3.6**	**57.4**
Soft clustering (# O-clusters = 10, # I-clusters = 1)						
$r_O = NA$	65.4±12.0	20.8±8.1	43.1	75.7±2.1	26.5±5.3	51.1
$r_O = 0.80, 0.90$	65.6±12.5	26.6±8.7	46.1	75.5±2.5	35.7±3.5	55.6
$r_O = 0.85, 0.95$	65.6±12.5	26.8±8.8	46.2	75.4±2.5	35.8±3.5	55.6
Using unlabeled dev set only						
Hard clustering (# O-clusters = 10, # I-clusters = 1)						
$r_O = NA$	65.6±11.1	22.3±9.6	43.9	75.1±2.5	28.6±5.3	51.8
$r_O = 0.80, 0.90$	63.1±10.4	20.9±6.3	42.0	71.2±3.0	26.6±2.5	48.9
$r_O = 0.85, 0.95$	65.9±12.7	28.1±8.5	47.0	75.7±2.3	38.4±3.5	57.1
Soft clustering (# O-clusters = 10, # I-clusters = 1)						
$r_O = NA$	65.5±11.1	22.2±9.5	43.9	75.1±2.5	28.5±5.2	51.8
$r_O = 0.80, 0.90$	66.3±11.4	25.4±8.4	45.8	75.1±2.3	35.8±4.0	55.4
$r_O = 0.85, 0.95$	66.3±11.5	25.5±8.7	45.9	75.1±2.3	36.0±4.1	55.6
Using the support sets only (# O-clusters = 10, # I-clusters = 1)						
One iteration	63.8±8.7	21.0±10.4	42.4	73.5±3.7	32.3±4.3	52.9

Table 3: Results for the domain adaptation experiments reported in F1-score. Results marked with † are taken from Das et al. (2022) and evaluation of our proposed method is done on the same 10 supports. Ratio constraints are noted in the order of datasets, i.e. $r_O = 0.80, 0.90$ means that the first dataset, CoNLL, is tested with a ratio of 0.80 and the second dataset, WNUT17, is tested with a ratio of 0.90.

Model	**1-shot**			**5-shot**		
	CoNLL	**WNUT17**	**Avg.**	**CoNLL**	**WNUT17**	**Avg.**
Proto †	49.9±8.6	17.4±4.9	33.7	61.3±9.1	22.8±4.5	42.1
NNShot †	61.2±10.4	22.7±7.4	41.9	74.1±2.3	27.3±5.4	50.7
StructShot †	62.4±10.5	24.2±8.0	43.3	74.8±2.4	30.4±6.5	52.6
CONTaiNER †	57.8±10.7	24.2±2.9	41.0	72.8±2.0	27.7±2.2	50.3
+ Viterbi †	61.2±10.7	27.5±1.9	44.4	**75.8**±**2.7**	32.5±3.8	54.2
K-Means with subspace selection						
Hard clustering (# O-clusters = 10, # I-clusters = 1)						
$r_O = NA$	65.4±12.0	20.9±8.2	43.2	75.7±2.1	26.5±5.3	51.1
$r_O = 0.80, 0.90$	62.6±10.6	23.6±7.1	43.1	71.9±3.0	29.3±3.1	50.6
$r_O = 0.85, 0.95$	**66.4**±**13.1**	**28.9**±**9.4**	**47.7**	**75.8**±**2.6**	**39.0**±**3.6**	**57.4**
K-Means without subspace selection						
Hard clustering (# O-clusters = 10, # I-clusters = 1)						
$r_O = NA$	64.0±14.5	21.7±7.4	42.9	72.9±2.9	22.1±4.9	47.5
$r_O = 0.80, 0.90$	60.6±14.4	25.9±6.4	43.3	70.5±4.8	29.3±5.0	49.9
$r_O = 0.85, 0.95$	65.2±14.7	28.2±9.1	46.7	72.5±3.2	38.3±2.3	55.4

Table 4: Ablation results. Constrained and unconstrained k-means, with and without the subspace selection step i.e. $\boldsymbol{U} = \boldsymbol{I}_d$.