Low-Rank Regularization for Sparse Conjunctive Feature Spaces:
An Application to Named Entity Classification

A. Primadhanty[1] X. Carreras[2] A. Quattoni[2]

[1]Universitat Politècnica de Catalunya

[2]Xerox Research Centre Europe

Challenge

Conjunction of sparse elementary features

↓

very sparse

Example: Named Entity Classification

A shipload of 12 tonnes of rice arrives in [*Umm Qasr port*] *in the Gulf*

$\phi_{\mathrm{l}}(l)$ $\phi_{\mathrm{e}}(e)$ $\phi_{\mathrm{r}}(r)$

↓ ↓ ↓

sparse sparse sparse

Approaches

ℓ_1 **or** ℓ_2

unseen conjunctions?

Contribution

Low-rank regularization for sparse conjunctive feature spaces

Propagate weight to unseen conjunctions

Learning algorithm

Convex relaxation of the low-rank minimization function

Experiments

Improvement over ℓ_1 & ℓ_2

Task

Given:

$x = \langle l, e, r \rangle$

Goal:

Classify x into one entity class y in the set $\mathcal{Y}$

A shipload of 12 tonnes of rice arrives in [*Umm Qasr port*] *in the Gulf*
l *e* *r*
↓
y?

Classifier

Log-Linear Model

$$\Pr(y \mid x; \theta) = \frac{\exp\{s_\theta(x, y)\}}{\sum_{y'} \exp\{s_\theta(x, y')\}}$$

$s_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is **scoring function** of entity tuples with a candidate class

θ are **parameters** of this function

Scoring Function

Feature-based linear model

$$s_\theta(x, y) = \phi(x) \cdot \mathbf{w}_y$$

$\phi : \mathcal{X} \to \{0, 1\}^n$ is a **feature function** representing entity tuples in an n-dimensional binary feature space

$\theta = \{\mathbf{w}_y\}_{y \in \mathcal{Y}}$ are **weight vector** for each class

Scoring Function

Left-right context model

$$s_\theta(\langle l, e, r \rangle, y) = \phi_{\mathrm{l}}(l)^\top \mathbf{W}_y \phi_{\mathrm{r}}(r)$$

$\phi_{\mathrm{l}} \in \mathbb{R}^{d_1}$ is a feature function representing **left contexts**

$\phi_{\mathrm{r}} \in \mathbb{R}^{d_2}$ is a feature function representing **right contexts**

$\mathbf{W}_y \in \mathbb{R}^{d_1 \times d_2}$ is **weight matrix** for each class, such that $\theta = \{\mathbf{W}_y\}_{y \in \mathcal{Y}}$

Low Rank Parameter Matrices

SVD

$$\mathbf{W}_y = \underbrace{\begin{bmatrix} u_{11} & \cdots & u_{1k} \\ \vdots & \vdots & \vdots \\ u_{d_1} & \cdots & u_{d_1 k} \end{bmatrix}}_{\mathbf{U}_y} \underbrace{\begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k \end{bmatrix}}_{\mathbf{\Sigma}_y} \underbrace{\begin{bmatrix} v_{11} & \cdots & \cdots & v_{1 d_2} \\ \vdots & \vdots & \vdots & \vdots \\ v_{k1} & \cdots & \cdots & v_{k d_2} \end{bmatrix}}_{\mathbf{V}_y^\top}$$

Consider that $\mathbf{W}_y$ has **rank** k

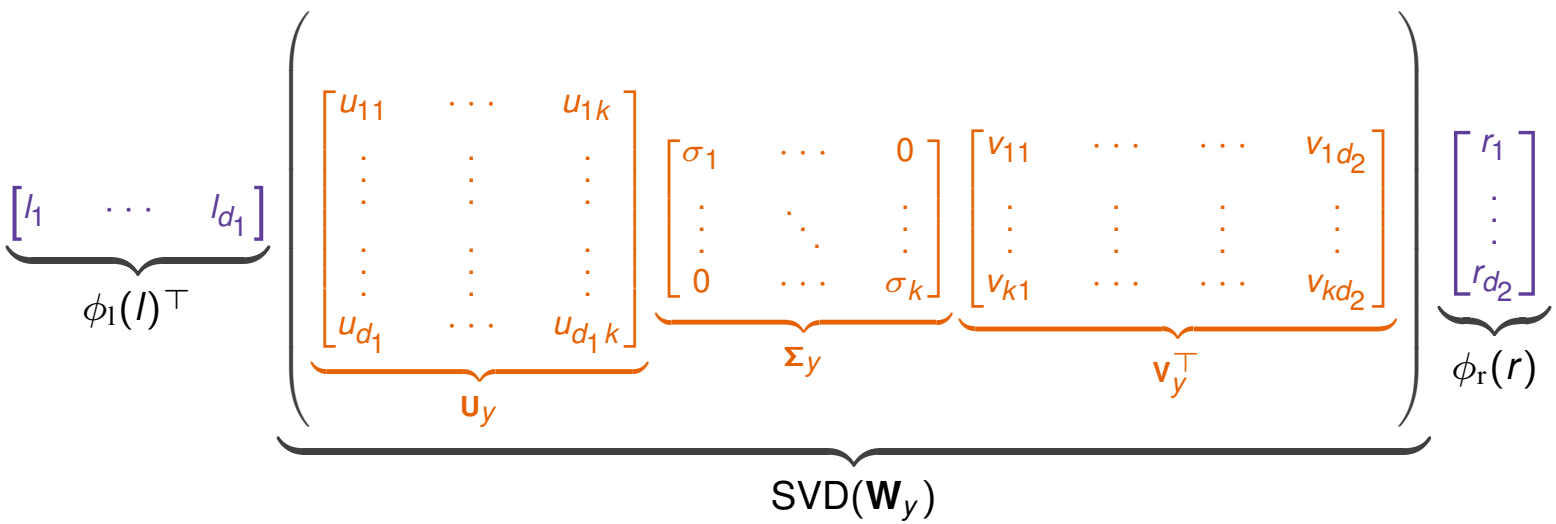
$\mathbf{U}_y \in \mathbb{R}^{d_1 \times k}$ and $\mathbf{V}_y \in \mathbb{R}^{d_2 \times k}$ are orthonormal projections

$\mathbf{\Sigma}_y \in \mathbb{R}^{k \times k}$ is a diagonal matrix of singular values

Score Function - Rewritten

Left-right context model

$$s_\theta(\langle l, e, r\rangle, y) = \phi_{\mathrm{l}}(l)^\top \mathbf{W}_y \phi_{\mathrm{r}}(r)$$



Score Function - Rewritten

Left-right context model

$$s_\theta(\langle l, e, r\rangle, y) = \phi_{\mathrm{l}}(l)^\top \mathbf{W}_y \phi_{\mathrm{r}}(r)$$

$$\underbrace{\left(\begin{bmatrix} l_1 & \cdots & l_{d_1} \end{bmatrix} \begin{bmatrix} u_{11} & \cdots & u_{1k} \\ \vdots & \vdots & \vdots \\ u_{d_1} & \cdots & u_{d_1 k} \end{bmatrix}\right)}_{\phi_{\mathrm{l}}(l)^\top \mathbf{U}_y} \underbrace{\begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k \end{bmatrix}}_{\mathbf{\Sigma}_y} \underbrace{\left(\begin{bmatrix} v_{11} & \cdots & \cdots & v_{1d_2} \\ \vdots & \vdots & \vdots & \vdots \\ v_{k1} & \cdots & \cdots & v_{kd_2} \end{bmatrix} \begin{bmatrix} r_1 \\ \vdots \\ r_{d_2} \end{bmatrix}\right)}_{\mathbf{V}_y^\top \phi_{\mathrm{r}}(r)}$$

Rank $k \rightarrow$ **intrinsic dimensionality** of the inner product behind the score function

Adding Entity Features

One parameter matrix per feature tag and class label, i.e. $\theta = \{\mathbf{W}_{t,y}\}_{t \in \mathcal{T}, y \in \mathcal{Y}}$

$$s_\theta(\langle l, e, r \rangle, y) = \sum_{t \in \phi_{\mathrm{e}}(e)} \phi_{\mathrm{l}}(l)^\top \mathbf{W}_{t,y} \phi_{\mathrm{r}}(r)$$

↓

Parameters: **tensor**

↓

Rank defined by *matricization*

Learning The Parameters

Objective Function

$$\underset{\mathbf{W}}{\operatorname{argmin}}\, L(\mathbf{W}) + \tau R(\mathbf{W})$$

$L(\mathbf{W})$ is a convex **loss function** (negative log-likelihood)

$R(\mathbf{W})$ is a **regularizer**

τ is a constant that trades off error and capacity

Minimizing rank $\rightarrow$ non-convex function

$\downarrow$

nuclear norm: convex relaxation

(Srebro & Shraibman, 2005)

Experimental Settings

Task	Named Entity Classification
Data	Annotated English CoNLL
Training	Minimal supervision (seeds) + large unlabeled data

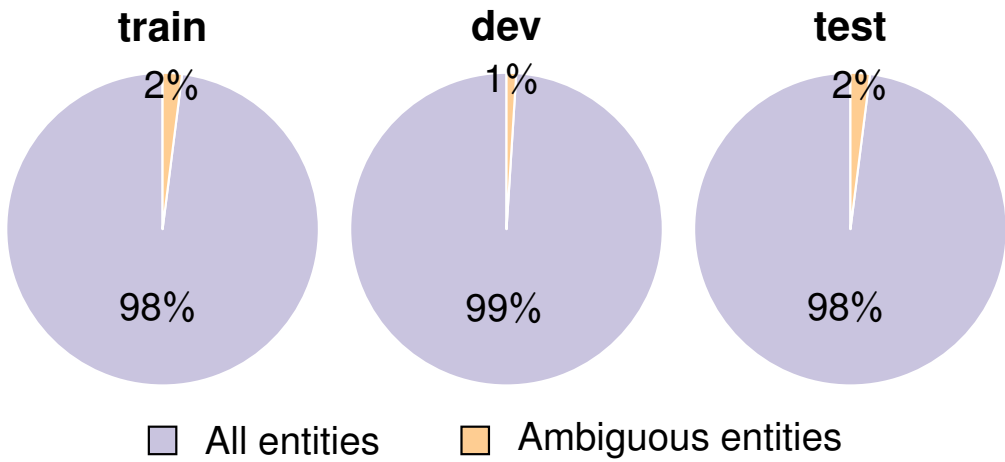
Class	10-30 Seed
PER	clinton, dole, arafat, yeltsin, wasim akram, lebed, dutroux, waqar younis, mushtaq ahmed, croft
LOC	u.s., england, germany, britain, australia, france, spain, pakistan, italy, china
ORG	reuters, u.n., oakland, puk, osce, cincinnati, eu, nato, ajax, honda
MISC	russian, german, british, french, dutch, english, israeli, european, iraqi, australian
O	year, percent, thursday, government, police, results, tuesday, soccer, president, monday, friday, people, minister, sunday, division, week, time, state, market, years, officials, group, company, saturday, match, at, world, home, august, standings

For each entity class, the seed of entities for the **10-30** set.

Experimental Settings

Task	Named Entity Classification
Data	Annotated English CoNLL
Training	Minimal supervision (seeds) + large unlabeled data
Evaluation	Mentions of unseen entities

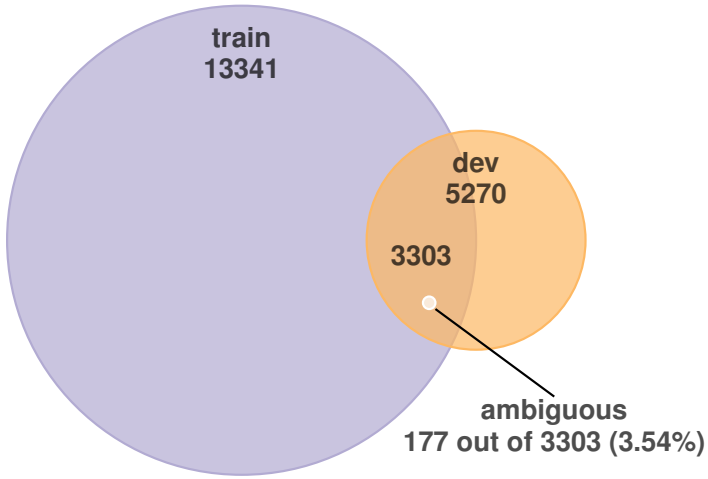
CoNLL 2003 English Corpus



Most entities in each set are non-ambiguous.

*Entities : unique candidate entities

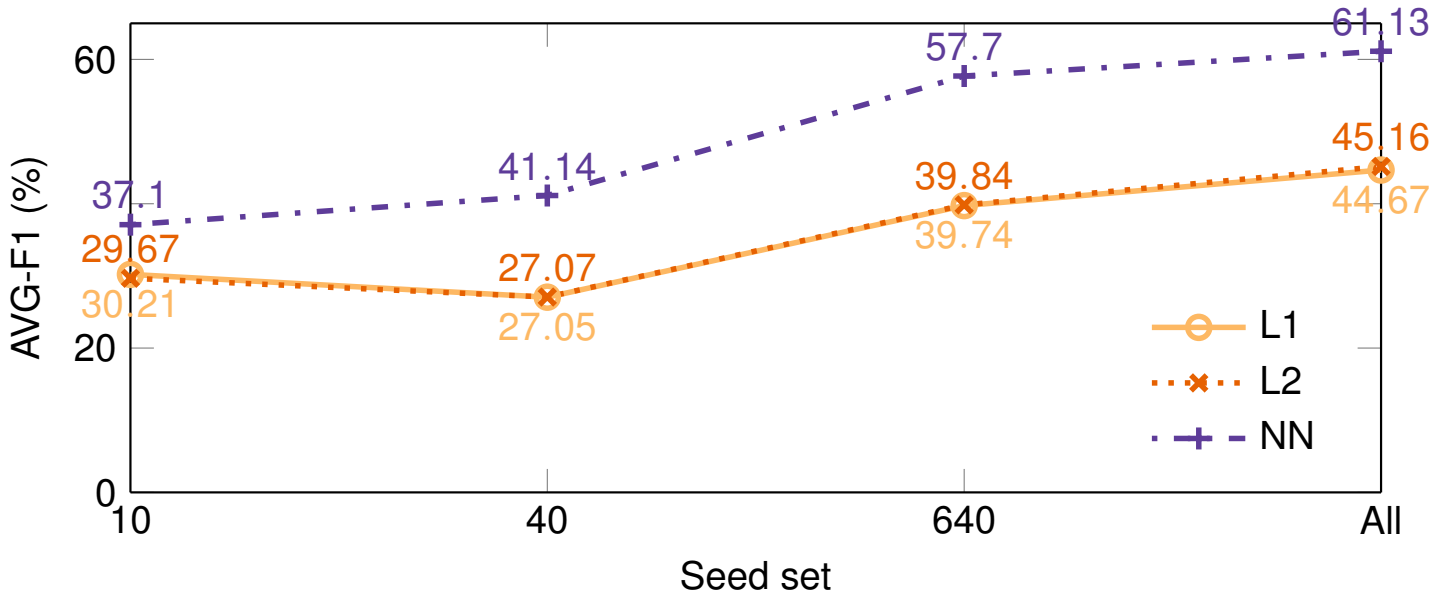
CoNLL 2003 English Corpus



Almost all seen entities that appear in dev can be directly classified as the same class.

*Entities : unique candidate entities

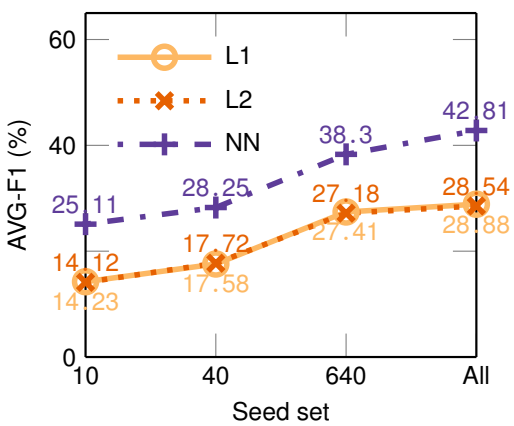
Results on dev set



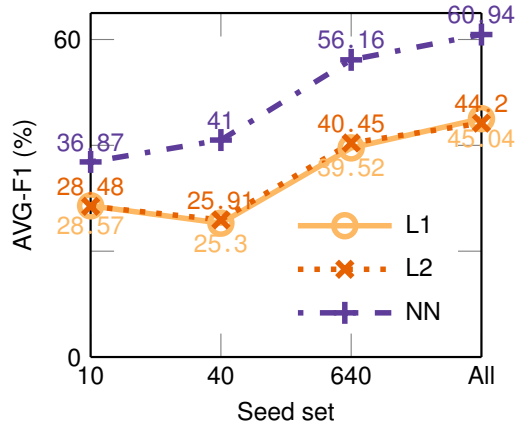
AVG-F1 on dev set using different seed set for training, comparing ℓ_1, ℓ_2 and nuclear-norm (NN) regularizer. Feature set: elementary features and all conjunctions of entity tags and left-right contexts (cluster & PoS), window size = 1

Seed set: number of examples per entity class (and 3$\times$ of non-entity examples)

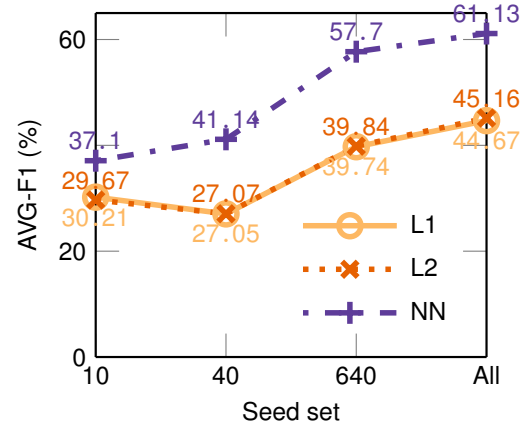
Results on dev set



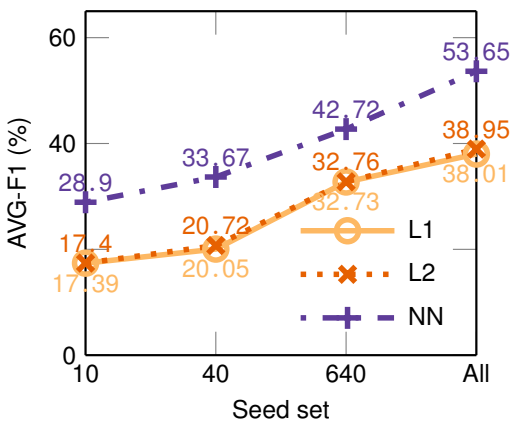
Only full conjunctions of left-right contexts (cluster), window size = 1



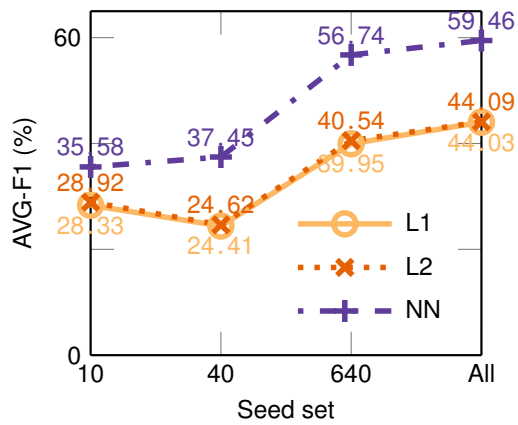
Elementary features and all conjunctions of entity tags and left-right contexts (cluster), window size = 1



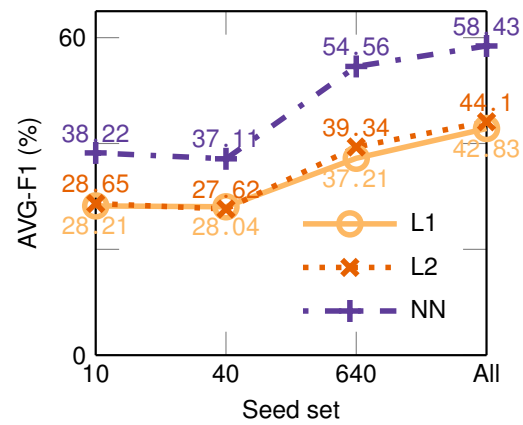
Elementary features and all conjunctions of entity tags and left-right contexts (cluster & PoS), window size = 1



Only full conjunctions of entity tags and left-right contexts (cluster), window size = 1

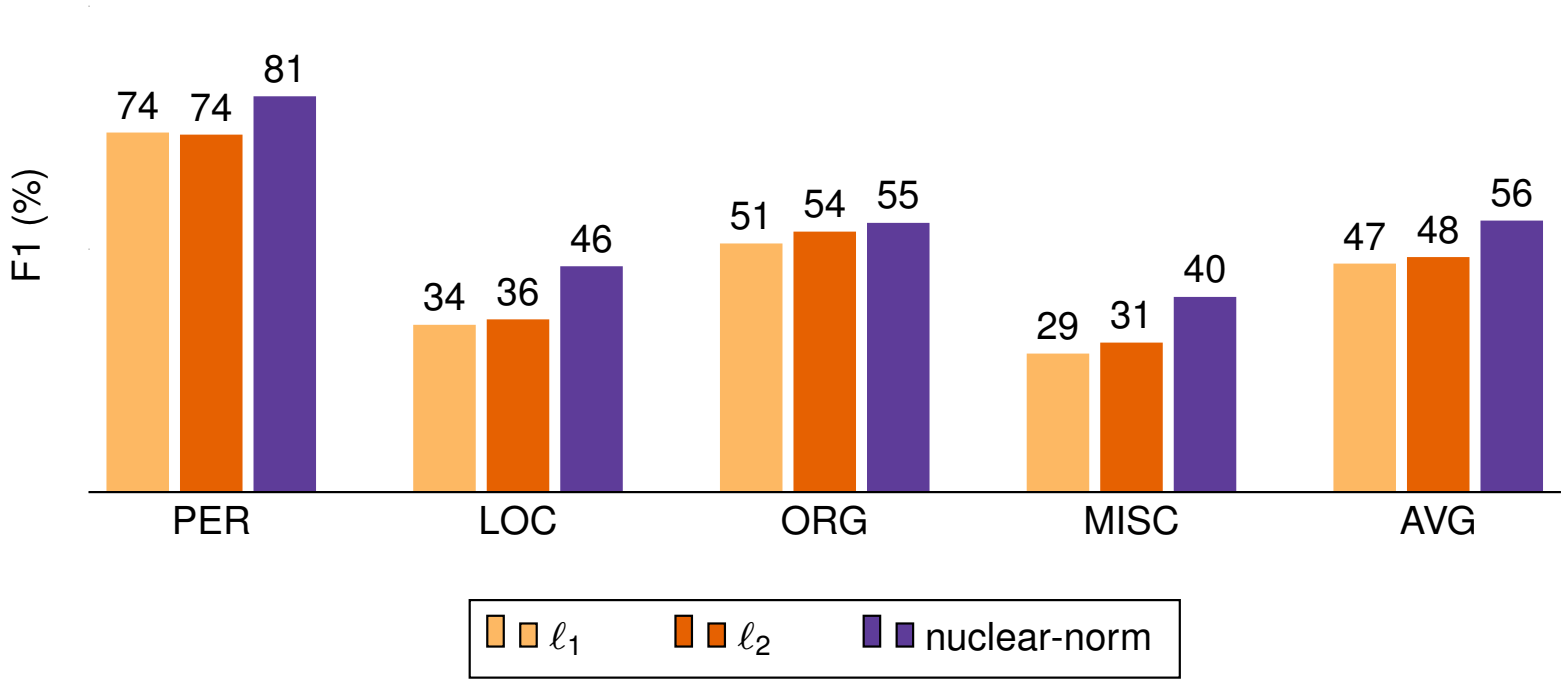


Elementary features and all conjunctions of entity tags and left-right contexts (cluster), window size = 2



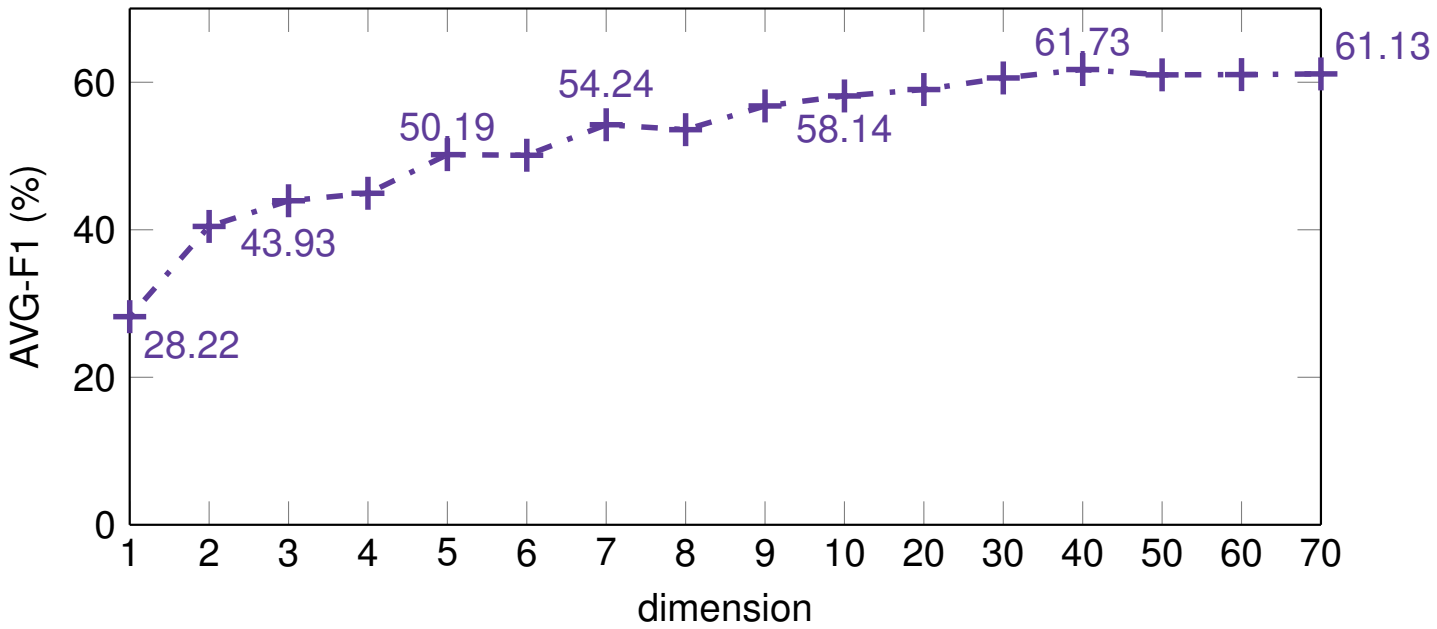
Elementary features and all conjunctions of entity tags and left-right contexts (cluster & PoS), window size = 2

Results on test set

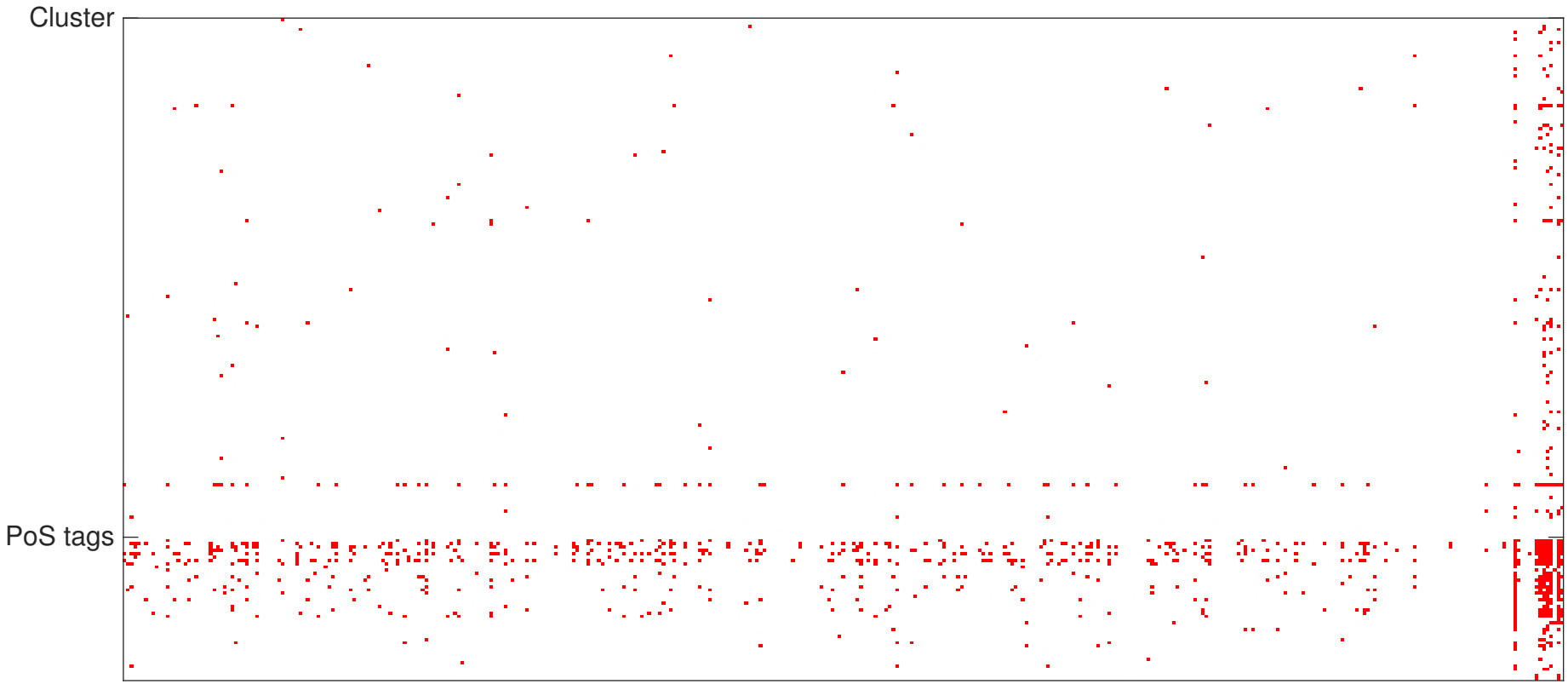


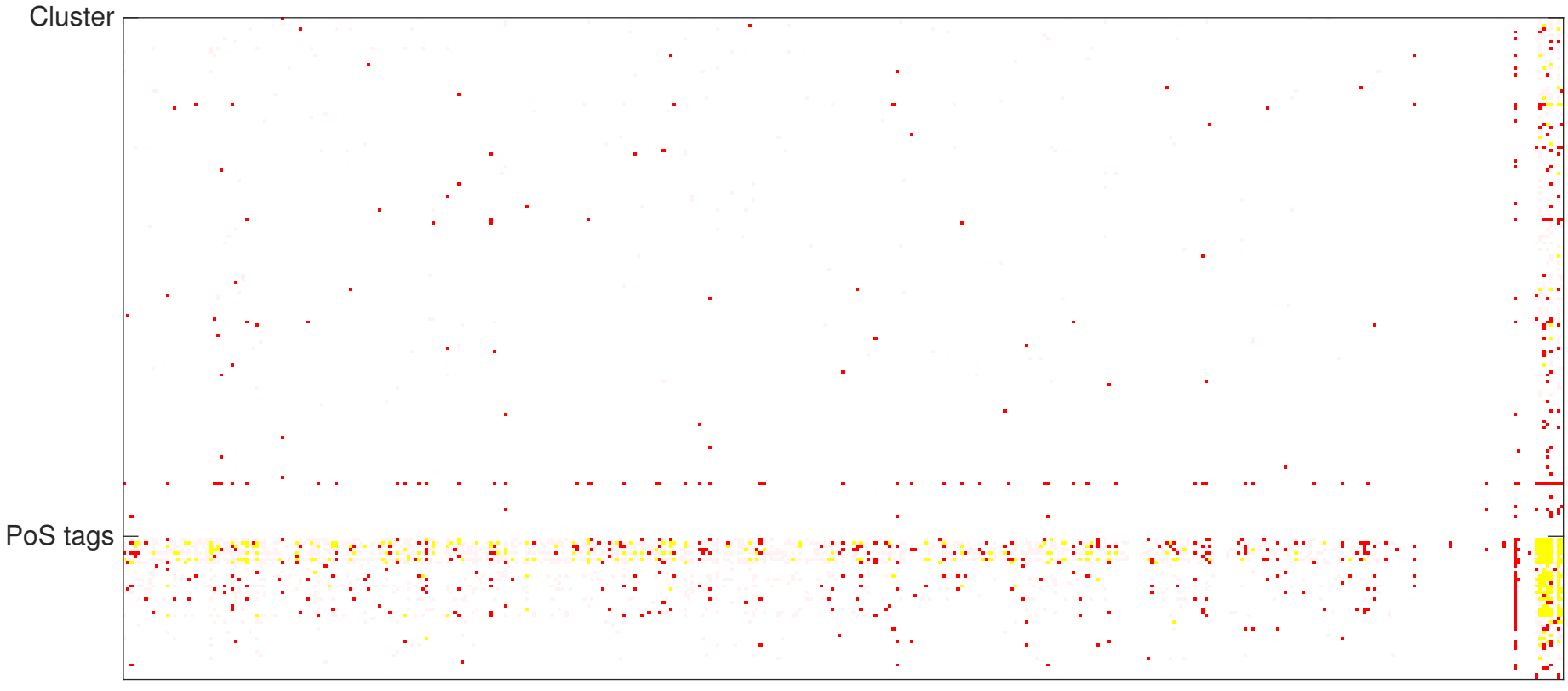
F1 performance on test set using “all” seed set for training, with best setting (based on results on dev) for each regularizers.

Model Dimensions



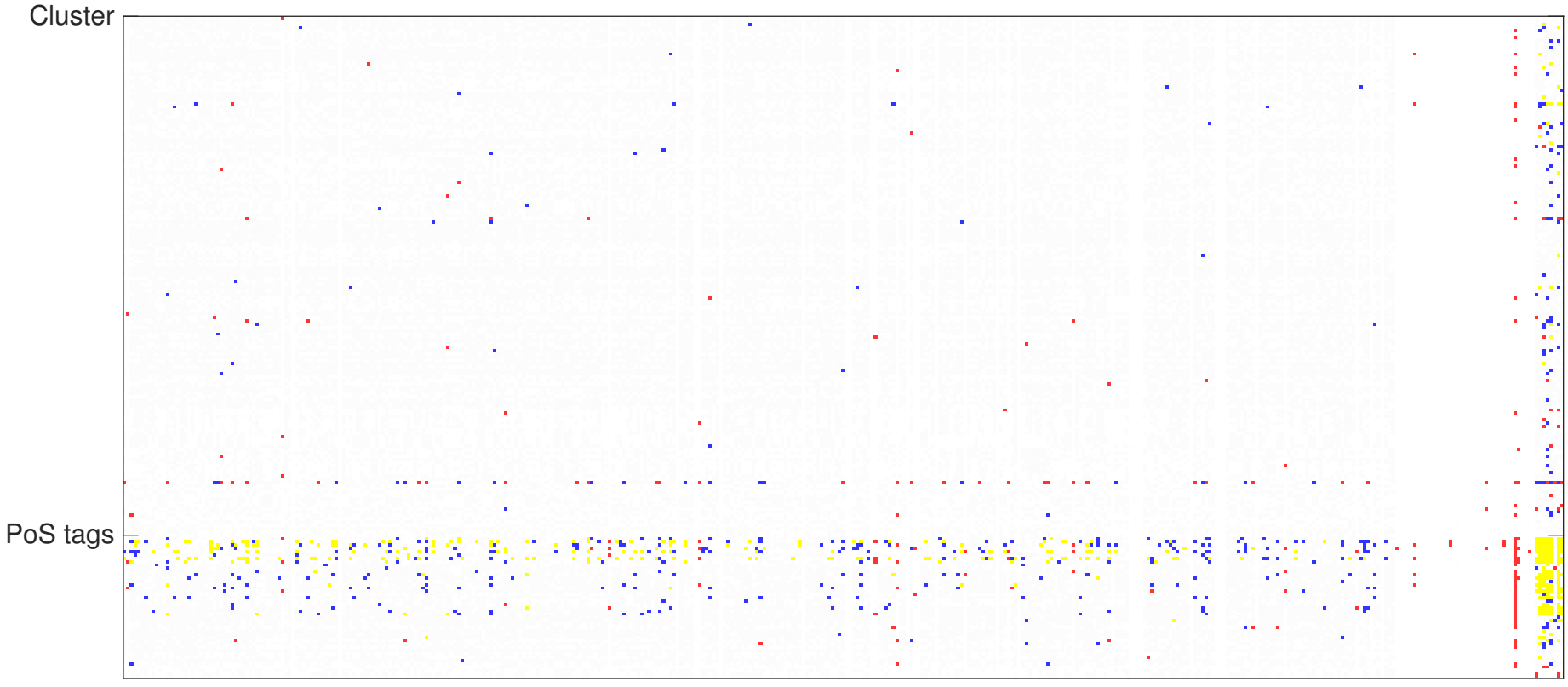
Avg. F1 on development for increasing dimensions, using the best low-rank model in development set trained with **all** seeds.


Cluster
PoS tags
Feature conjunctions in dev set
conjunctions in dev



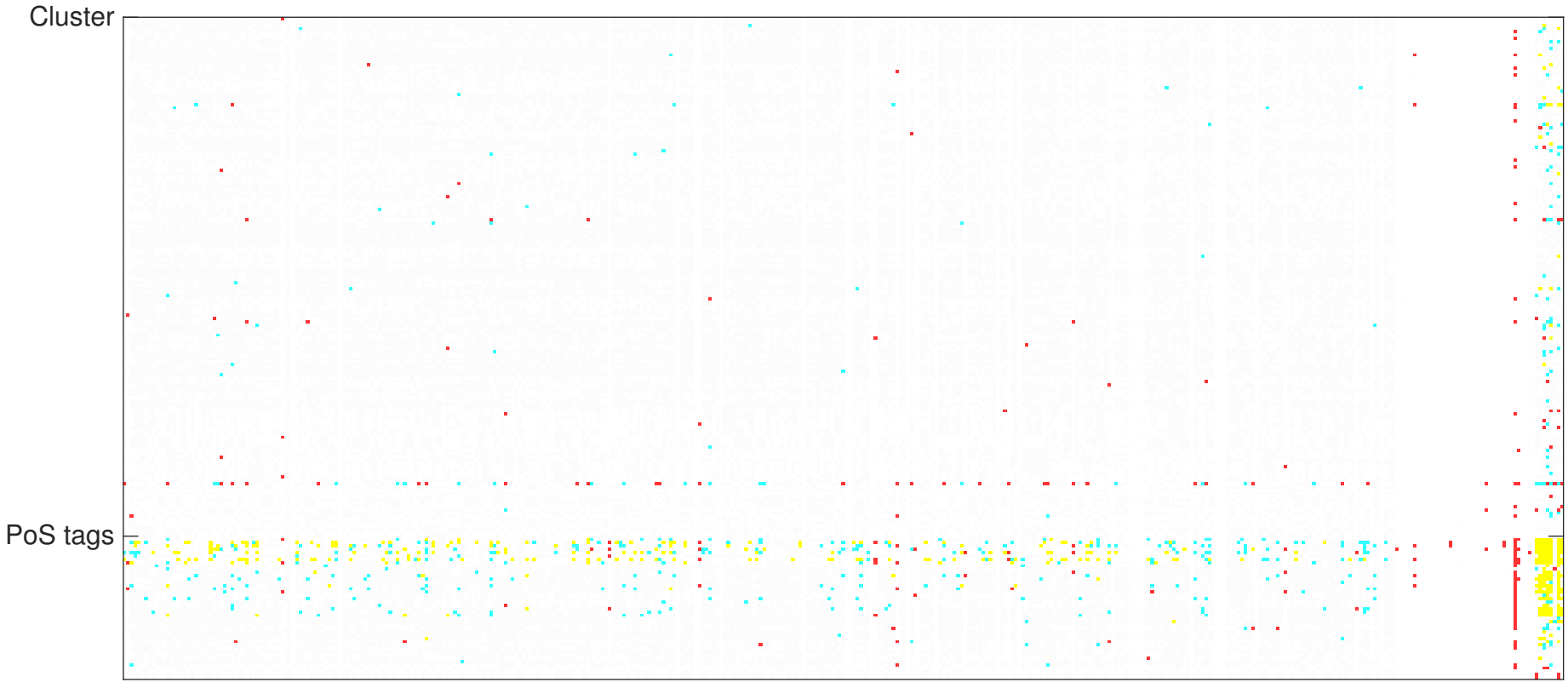
Feature conjunctions in dev set

conjunctions in dev that are **unseen** in train (with 10 seeds)
conjunctions in dev that are **seen** in train (with 10 seeds)



Feature conjunctions in dev set

■ conjunctions in dev that are **unseen** in train (with 10 seeds) and has **zero weight**
■ conjunctions in dev that are **seen** in train (with 10 seeds)
■ conjunctions in dev that are **unseen** in train but assigned **non-zero weight** by model trained on 10 seeds



Feature conjunctions in dev set

conjunctions in dev that are **unseen** in train (with 10 seeds) and has **zero weight**

conjunctions in dev that are **seen** in train (with 10 seeds)

conjunctions in dev that are **unseen** in train but assigned **non-zero weight** by model trained on 10 seeds

Conclusion

Low-rank regularization framework for sparse conjunctive feature spaces

Tensors

Nuclear-norm

Experimented on learning entity classifiers

Compare to ℓ_1 and ℓ_2 penalties $\rightarrow$ better results

Illustrated weight propagation to unseen conjunctions

Future works : explore different tensor transformations

Thank you!