Group-Sparse Matrix Factorization for Transfer Learning of Word Embeddings

Kan Xu

University of Pennsylvania

Joint work with Xuanyi Zhao, Hamsa Bastani, and Osbert Bastani

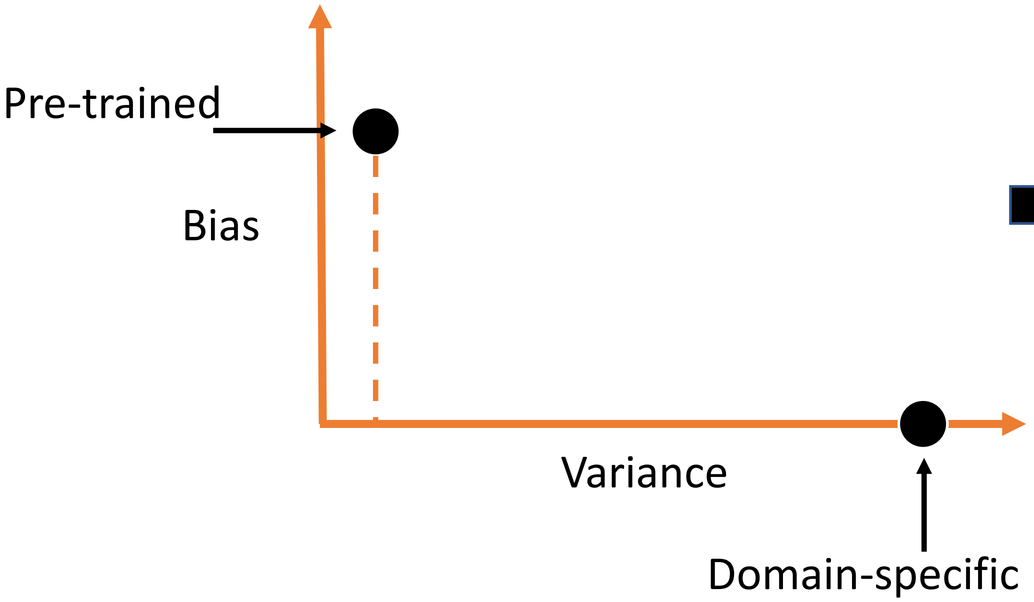
Motivation of This Research

- **Any drawbacks of the current word embedding models (e.g.** GloVe [Pennington, Socher, et al. (2014)], Word2Vec [Mikolov, Sutskever, et al. (2013)], etc.**)?**
- Domain scarcity
  - ML methods need big data (e.g. Wikipedia + Gigaword, 6B tokens, 400K vocab)
  - Domain text data might be small (e.g. rare disease, new product)
- Domain difference
  - Different meanings in a target domain (context)
  - Difference might be large
  - E.g. '**positive**' in its common meaning VS. as a medical condition in nurses' notes: **opposite sentiments!**

	Common	**Healthcare**
Sentiment	Positive	Negative

Bias-Variance Tradeoff

- Using **pre-trained embeddings** from Wiki: (*source domain*)
  - biased but low variance
- Using **domain-specific embeddings**: (*target domain*)
  - unbiased but high variance



Transfer Learning (Domain Adaptation):
pre-trained embeddings + domain text data

Previous Results

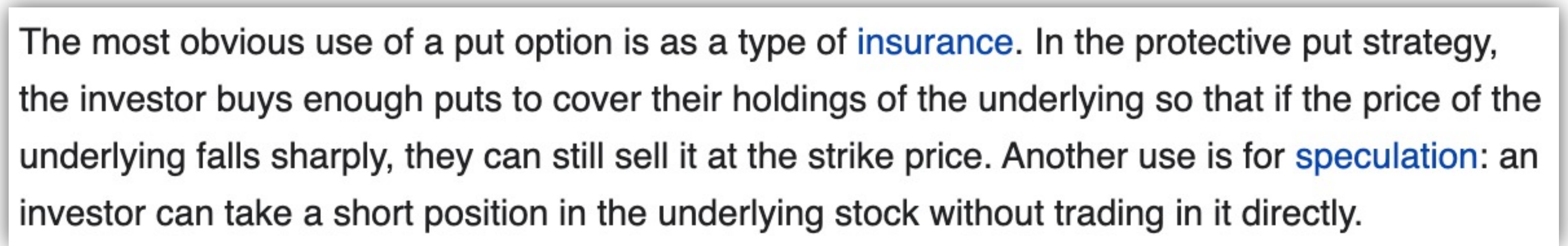
- Fine-tune Pre-trained Embeddings:
  - Mittens [Dingwall & Potts (2018)]: ℓ_2 regularization
  - CCA/KCCA [Sarma, Liang, et al. (2018)]: weighted sum of aligned embeddings
  - …

No Theoretical Guarantee!

- Our approach:
  - With theoretical guarantee
  - Matrix factorization [Ge, Jin, et al. (2017), Negahban & Wainwright (2011)]
  - Group sparsity [Lounici, Pontil, et al. (2011)]
  - **Interpretable & outperforms fine-tuning**

Transfer Learning

- Insights
  - **Many words** keep the same meaning across contexts
  - Words with a different meaning in a target domain are '**sparse**'

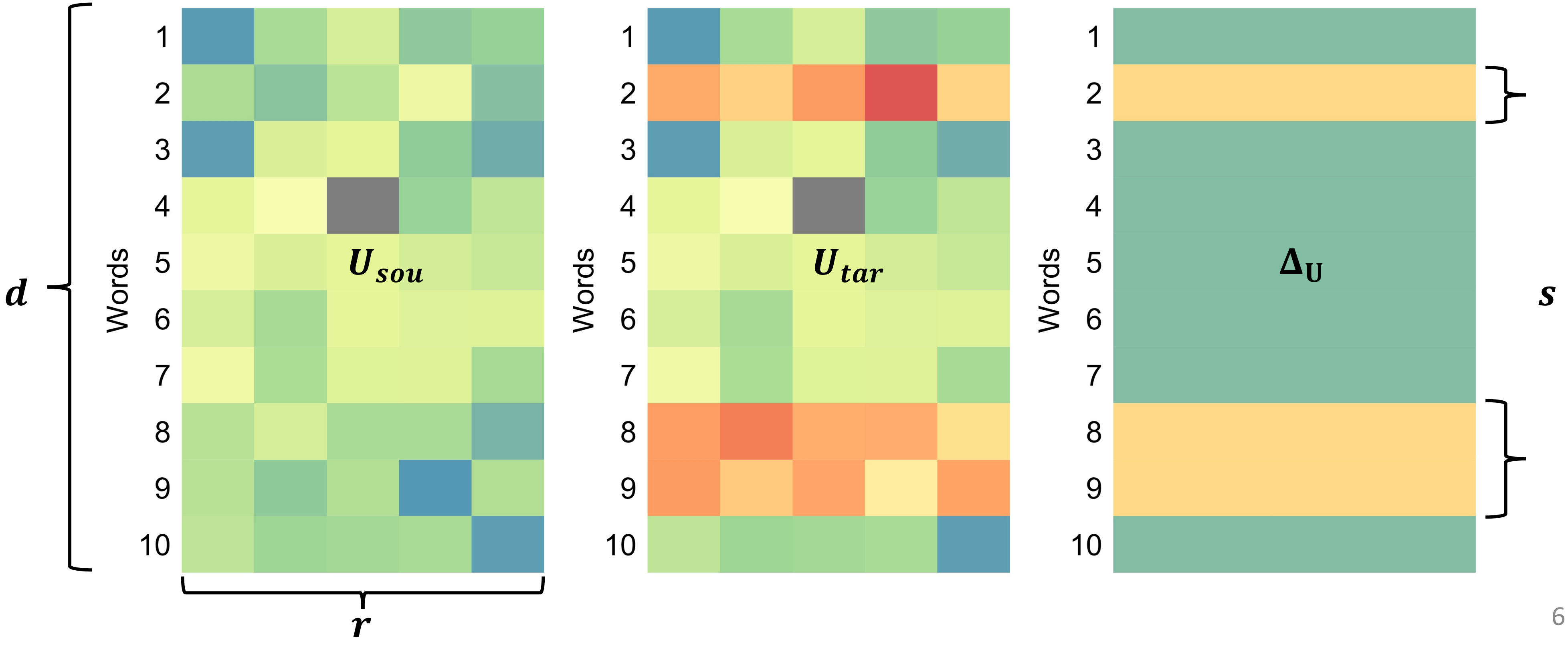


The most obvious use of a put option is as a type of insurance. In the protective put strategy, the investor buys enough puts to cover their holdings of the underlying so that if the price of the underlying falls sharply, they can still sell it at the strike price. Another use is for speculation: an investor can take a short position in the underlying stock without trading in it directly.

  - e.g. a paragraph from the article 'put option' on Wikipedia
    - 'put', 'option', 'strike', 'speculation', 'short', 'stock' have specific meanings in finance
    - The proportion of these words is $\frac{6}{71} \approx \mathbf{8.5\%}$

Transfer Learning

- **Sparsity** of words → **sparsity** of word embedding matrix
- Toy example
  - 10 words with word embedding of dimension 5
  - Word 2, 8, 9 have domain-specific meanings
  - Word embedding matrix difference is **row-sparse**



Transfer Learning

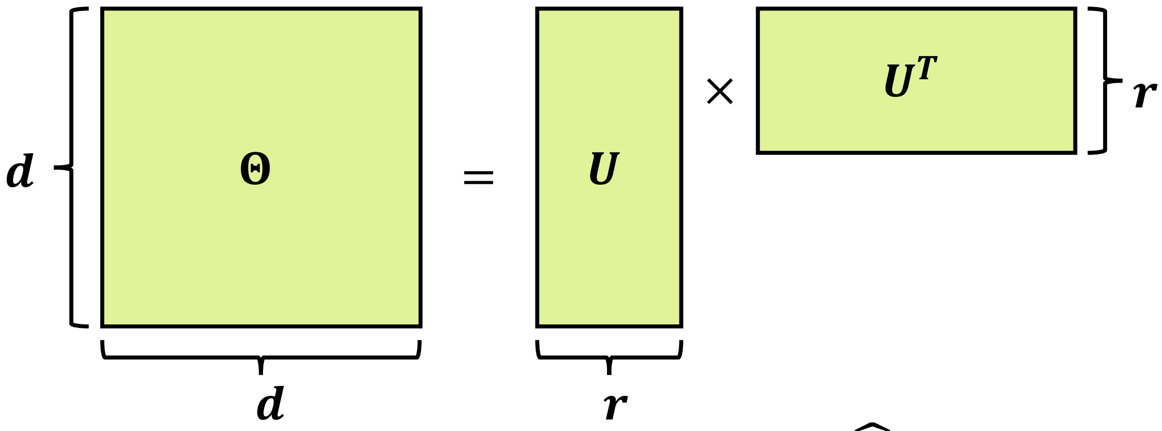
- Main Idea
  - **Target domain texts + pre-trained embedding**: row-sparse deviation
  - **Group-sparse penalty** on Δ_U via $\ell_{2,1}$ norm, i.e.

$$\| \Delta_U \|_{2,1} = \sum_{i=1}^{d} \| \Delta_U^i \|_2$$

  - Each row of Δ_U, i.e. Δ_U^i, is a group
  - Unsupervised VS. supervised (group Lasso)

- Method: a **two-stage** estimator
  - **Stage 1**: Obtain $\widehat{\boldsymbol{U}}_{sou}$ (or take the pretrained embeddings available)
  - **Stage 2**: Estimate $\boldsymbol{U}_{tar}$, regularizing $\boldsymbol{U}_{tar}$ towards $\widehat{\boldsymbol{U}}_{sou}$ via $\ell_{2,1}$ norm

Problem Formulation



- **Source domain**: $X_s = \mathcal{A}_s(\Theta_s) + \epsilon_s$
- **Target domain**: $X_t = \mathcal{A}_t(\Theta_t) + \epsilon_t$
    - X's – e.g. word co-occurrence
    - Θ's – e.g. true co-occurrence prob

- $\hat{U}_s R$ is also a solution!
    - R is orthogonal matrix
- **Estimation error of word embedding:**
    - $\| \hat{U}_t - U_t R_{(\hat{U}_t, U_t)} \|_{2,1}$

- **Two-stage estimator:**
- **Stage 1**: $\min_{U_s} \frac{1}{n_s} \| X_s - \mathcal{A}_s\left(U_s U_s^T\right) \|_2^2$
- **Stage 2**: $\min_{U_t} \frac{1}{n_t} \| X_t - \mathcal{A}_t\left(U_t U_t^T\right) \|_2^2 + \lambda \| U_t - \hat{U}_s \|_{2,1}$

Theoretical Results

- “**Target-scarce and source-rich**” regime:
  - $n_{sou} \gg d^2$, $n_{tar} \ll d^2$, $s \ll d$
- **Can we improve the estimation error of $\boldsymbol{U_{tar}}$?**

Only Source:

$$\mathcal{O}(\| \Delta_U \|_{2,1} + \sqrt{\frac{d^2}{n_s}})$$

Only Target:

$$\mathcal{O}(\sqrt{\frac{d^2}{n_t}})$$

Combine both?

Theoretical Results

- **Debias** source-domain estimator
- Require **exponentially fewer** target-domain data

Only Source:

$$\mathcal{O}(\| \Delta_U \|_{2,1} + \sqrt{\frac{d^2}{n_s}})$$

Only Target:

$$\mathcal{O}(\sqrt{\frac{d^2}{n_t}})$$

Group sparsity helps! →

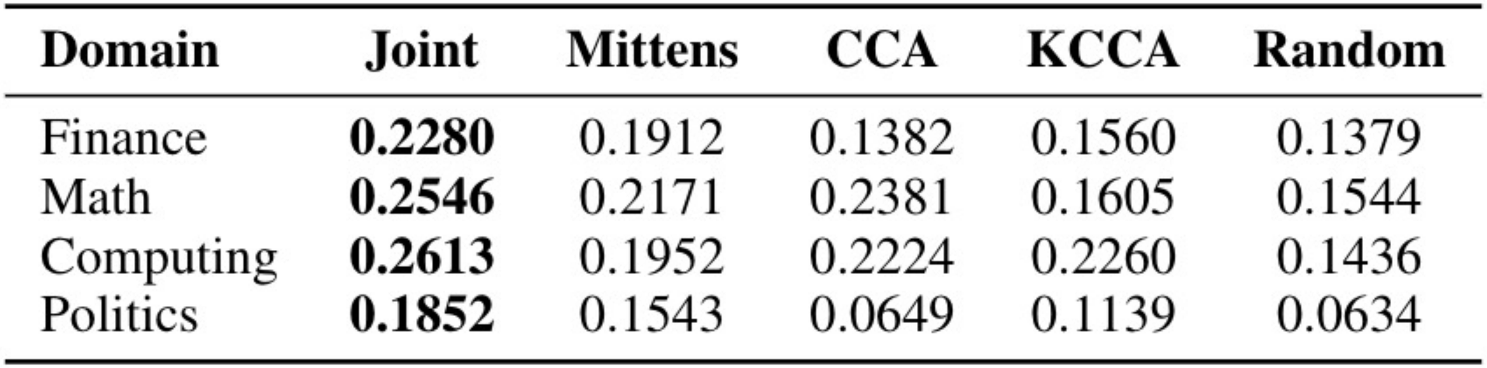
$$\mathcal{O}(\sqrt{\frac{s^2 \log(d)}{n_t}} + \sqrt{\frac{d^2}{n_s}})$$

Empirical Results

- Interpretability
    - Single domain-specific Wiki articles
- Efficiency in downstream prediction
    - Clinical trial eligibility data

Interpretability

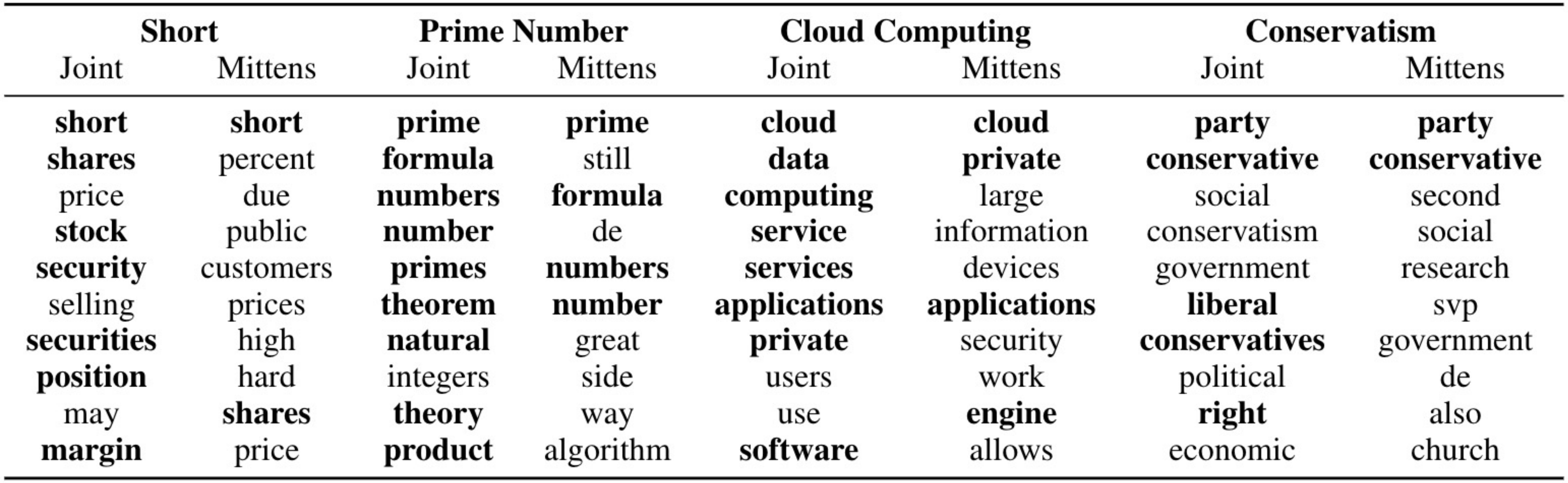
- **How accurate to identify domain words?**
  - Score words by the Euclidean distance of estimators with pre-trained embeddings (GloVe): **higher** → **more likely** to be domain words
  - Calculate F1-score: selecting the top 10% as domain words
  - Random: words picked randomly



Domain	Joint	Mittens	CCA	KCCA	Random
Finance	**0.2280**	0.1912	0.1382	0.1560	0.1379
Math	**0.2546**	0.2171	0.2381	0.1605	0.1544
Computing	**0.2613**	0.1952	0.2224	0.2260	0.1436
Politics	**0.1852**	0.1543	0.0649	0.1139	0.0634

Interpretability

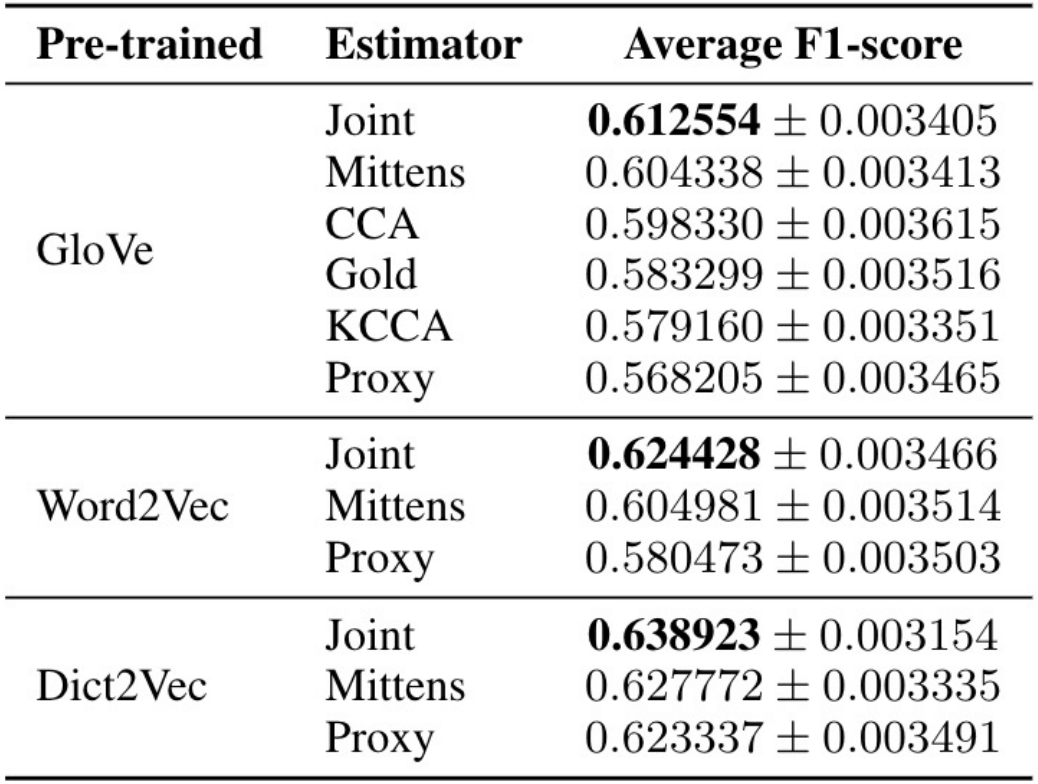
- Top 10 words selected by our estimator and Mittens



Short		Prime Number		Cloud Computing		Conservatism	
Joint	Mittens	Joint	Mittens	Joint	Mittens	Joint	Mittens
short	**short**	**prime**	**prime**	**cloud**	**cloud**	**party**	**party**
shares	percent	**formula**	still	**data**	**private**	**conservative**	**conservative**
price	due	**numbers**	**formula**	**computing**	large	social	second
stock	public	**number**	de	**service**	information	conservatism	social
security	customers	**primes**	**numbers**	**services**	devices	government	research
selling	prices	**theorem**	**number**	**applications**	**applications**	**liberal**	svp
securities	high	**natural**	great	**private**	security	**conservatives**	government
position	hard	integers	side	users	work	political	de
may	**shares**	**theory**	way	use	**engine**	**right**	also
margin	price	**product**	algorithm	**software**	allows	economic	church

Downstream Prediction

- Short clinical statements → eligibility for cancer clinical trials
- Logistic regression with ℓ_2 penalty
- Dict2Vec [Tissier, Gravier, et al. (2017)] : Word2Vec + dictionary definition



Pre-trained	Estimator	Average F1-score
GloVe	Joint	**0.612554** ± 0.003405
	Mittens	0.604338 ± 0.003413
	CCA	0.598330 ± 0.003615
	Gold	0.583299 ± 0.003516
	KCCA	0.579160 ± 0.003351
	Proxy	0.568205 ± 0.003465
Word2Vec	Joint	**0.624428** ± 0.003466
	Mittens	0.604981 ± 0.003514
	Proxy	0.580473 ± 0.003503
Dict2Vec	Joint	**0.638923** ± 0.003154
	Mittens	0.627772 ± 0.003335
	Proxy	0.623337 ± 0.003491

Note: Gold – domain embedding; Proxy – pretrained embedding

Conclusion

- Introduced **a two-stage estimator** to learn domain word embeddings
- Required **exponentially fewer** domain textual data
- First paper to **theoretically** justify the efficiency of transfer learning in word embedding
- Our estimator is **interpretable** and **efficient in downstream prediction**

Thank you!

kanxu@sas.upenn.edu

Questions and comments are welcome!