Deep Text Mining of Instagram Data Without Strong Supervision

Master's Thesis Defense

Kim Hammar

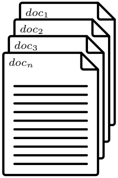
KTH Royal Institute of Technology

kimham@kth.se

June 1, 2018

Extracting information and detecting patterns in unstructured text

Corpora



Analysis

Donald J. Trump
@realDonaldTrump

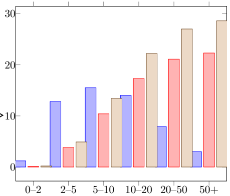
Despite the constant negative press $\overbrace{\text{covfefe}}^{\text{trend}}$

AFTONBLADET

$\overbrace{\text{GW}}^{person}$ om utpekade mannen: Lite för $\overbrace{\text{virrig}}^{adjective}$ och konstig. Leif GW Persson har tagit del av de nya uppgifterna om $\overbrace{\text{Palmemordet}}^{event}$. Enligt kriminologen är ”Skandiamannen” nu polisens huvudspår. – Det är jag ganska säker på. Men jag tycker han verkar lite för virrig och konstig för att stämma, säger han

Outfit 1 or 2? Talking $\overbrace{\text{blazers}}^{\text{clothing item}}$ on the blog today

Patterns



Text understanding is hard → AI-Complete Problem

Challenge: Text Ambiguity

- **Ambiguous sentences: we need context to understand**

  Did you see her dress?

Challenge: Text Ambiguity

- **Ambiguous sentences: we need context to understand**

Did you see her dress?

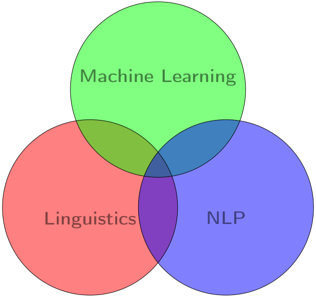
Yes I was in the hall with her when she dressed.
Yes It was gorgeous!

Text Mining Methods

- **Classic NLP & Text Mining: Linguistic Rules**
  - *"Two capital words in a row"* $\rightarrow$ A person's name
- **Data Driven Text Mining: Machine Learning Models**

$$\nabla_\theta L = \left(\frac{\partial L}{\partial W_{1,1}}, \ldots, \frac{\partial L}{\partial W_{n,n}}\right)$$

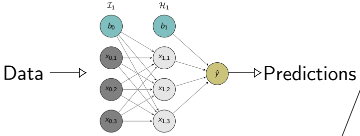
- Data $\Longrightarrow$ Model θ $\Longrightarrow$ Loss $L(\theta)$ $\Longrightarrow$ Outputs



Unsupervised Text Mining in Social Media

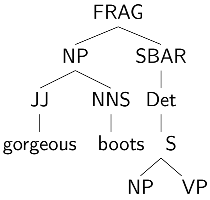
Unannotated Data

Machine Learning



Model

Natural Language Processing (NLP)



"Noisy" Text

outfit of the day #goals

Big Data

40B images

500M daily activity

Domain Under Study: Fashion

Automatic predictions:

Items: jeansshorts, ...

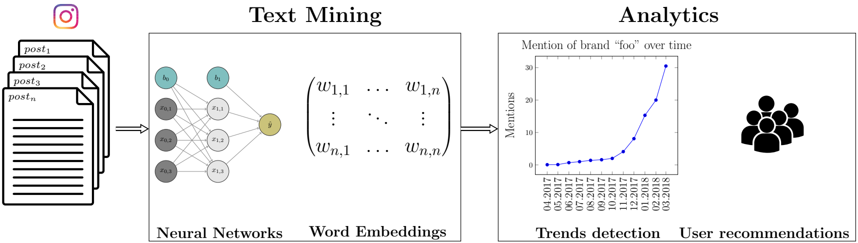
fabrics: denim,...

style: casual, ..

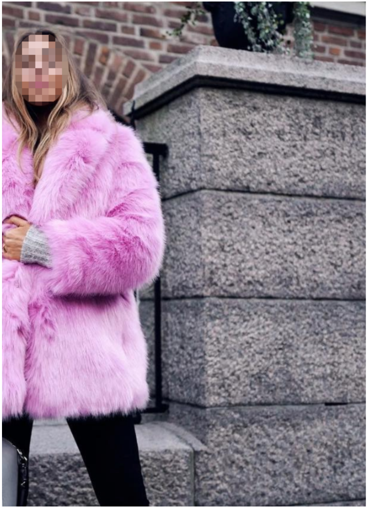
⋮

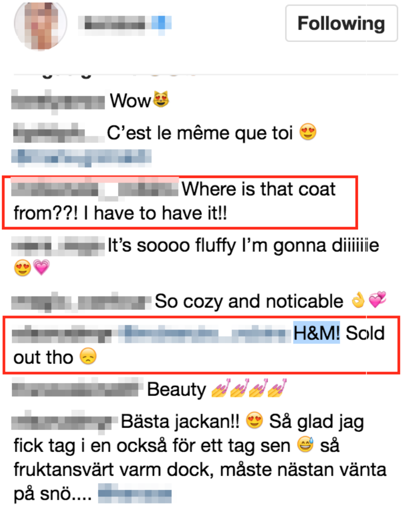
Problem: Unsupervised Text Mining from Instagram

- **Input:** Noisy Text (Image Captions, User Comments)
- **Output:** Fashion Attributes (Items, Fabrics, Brands... etc.)

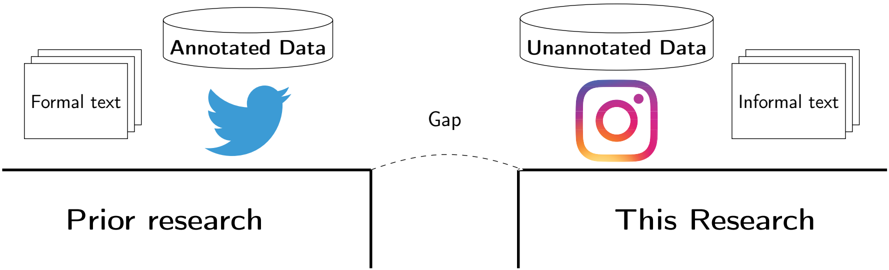


Example Instagram Post





How Our Research Stands Out From Prior Work



Contributions & Roadmap

1. **An empirical study of Instagram text**
   - No previous study on Instagram text exists that we are aware of
2. **Unsupervised extraction of fashion attributes from Instagram using Word Embeddings**
   - The first evaluation of word embeddings for Instagram
   - The first distributed implementation of the FastText algorithm
   - We confirm prior results on IE and apply it to a new domain
3. **Novel pipeline for classification with weak supervision & deep learning**
   - Extension of the data programming paradigm to the multi-label setting

[0] All code and most of the data is open source: https://github.com/shatha2014/FashionRec

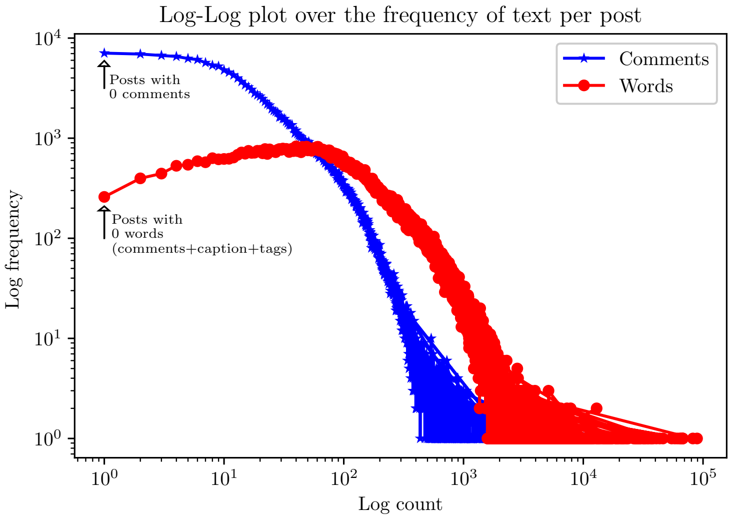
Instagram Text is Noisy and Multi-Lingual

A case study of a corpora with 143 fashion accounts, 200K posts, 9M comments

- **Instagram text is noisy**: 47% OOV words when including URLs, emojis etc. Otherwise 30% (compared to 25% on Twitter)
- **Comment sections are multi-lingual**: All accounts are English, still only 52% of comments are English (total 97 languages identified)
- **The text is ungrammatical**: Informal spelling, unreliable capitalization.

Instagram Text Distribution Has a Long Tail

A case study of a corpora with 143 fashion accounts, 200K posts, 9M comments

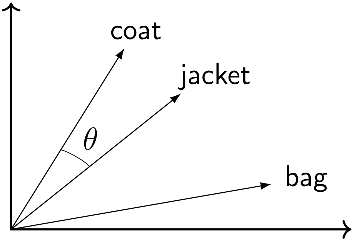


Conclusion from Case Study: Prefer Statistical Methods Rather Than Symbolic NLP Methods

- **Instagram Text is noisy, multi-lingual, and un-grammatical**
  - $\rightarrow$ Linguistic methods for text mining are fragile
  - $\rightarrow$ Syntactic text matching is difficult (many languages, many synonyms, online-specific tokens etc)
- **We propose: Word Embeddings as a key component in information extraction from Social Media**
  - Demonstrated in the second contribution of the thesis

Word Embeddings Are Distributed Representation of Words

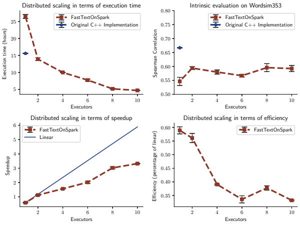
- Word Embeddings are vectors in $\mathbb{R}^d$, $\quad d \approx 300$
- **Derived with optimization using the *Distributional Hypothesis*[1]**
  - $\rightarrow$ Words that occur in similar contexts will obtain similar vectors
  - *"You shall know a word by the company it keeps"* - Firth '57[2]
  - $\rightarrow$ we can use the word knowledge in word embeddings for IE

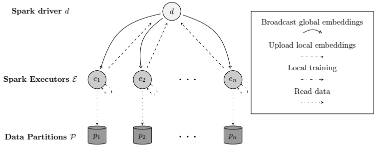


[1] Zellig S Harris. "Distributional structure". In: *Word* 10.2-3 (1954), pp. 146–162.

[2] J. R. Firth. "A synopsis of linguistic theory 1930-55." In: 1952-59 (1957), pp. 1–32.

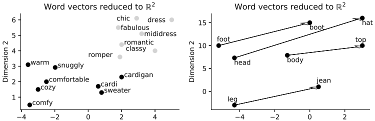
Contributions On Word Embeddings

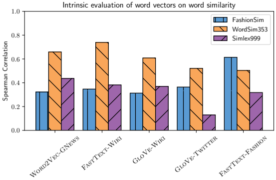


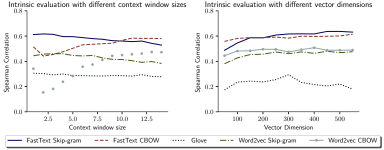


FastTextOnSpark:

- A Scalable Implementation of FastText







Instagram Word Embeddings:

- Hyperparameter Tuning
- Comparison With Off-The-Shelf Embeddings
- Extrinsinc and Intrinsic Evalutations

Unsupervised Information Extraction using Word Embeddings and a Fashion Ontology

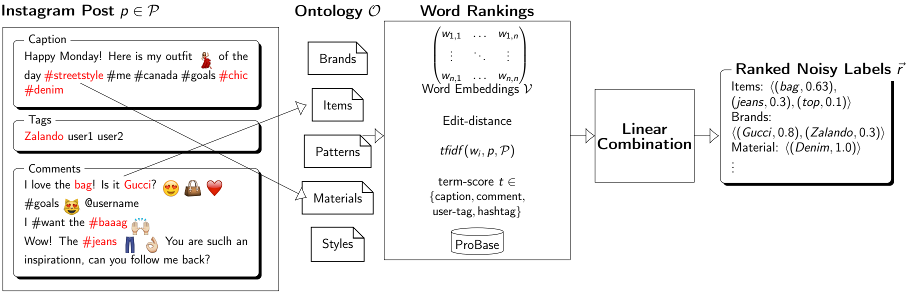


Figure: An information extraction system for social media text. The system extracts fashion details from text associated with Instagram posts.

Word Embeddings Outperform Syntactic Baseline for IE

- **IE with word embeddings outperform IE based on edit-distance ($p < 0.05$)**

Method/Category	*NDGC@1*	*NDGC@3*	*NDGC@5*	*NDGC@10*	*P@1*	*P@3*	*P@5*	*P@10*	*MAP*
SemCluster/Item	**0.833**	**0.658**	**0.691**	**0.807**	**0.833**	**0.546**	**0.454**	**0.309**	**0.733**
SynCluster/Item	0.781	0.581^{-}	0.607^{-}	0.767^{-}	0.781	0.474^{-}	0.370^{-}	0.296	0.641^{-}
SemCluster/Style	**0.399**	**0.505**	**0.519**	**0.548**	**0.417**	**0.204**	**0.139**	**0.069**	**0.539**
SynCluster/Style	0.367	0.415^{-}	0.425^{-}	0.507	0.367	0.130^{-}	0.123	0.069	0.474^{-}
SemCluster/Pattern	0.087	0.179	0.353	0.444	0.087	0.110	0.169	**0.118**	0.296
SynCluster/Pattern	**0.108**	**0.413**	**0.498**	**0.512**	**0.108**	**0.221**	**0.193**	0.117	**0.395**
SemCluster/Material	**0.296**	**0.286**	**0.324**	**0.393**	**0.286**	**0.264**	**0.233**	**0.165**	**0.373**
SynCluster/Material	0.113^{-}	0.104^{-}	0.137^{-}	0.209^{-}	0.113^{-}	0.107^{-}	0.109^{-}	0.092^{-}	0.227^{-}
SemCluster/Brand	**0.062**	**0.066**	**0.062**	**0.064**	**0.032**	**0.056**	**0.036**	**0.039**	**0.194**
SynCluster/Brand	0.016	0.010	0.010	0.010	0.016	0.005	0.003	0.002	0.159

[2] A smaller, hand-labeled dataset by experts was used for evaluation

Another Approach: Deep Learning with Weak Supervision

Text Mining with Word Embeddings and a smaller domain ontology:

Pros:

- Does not require annotated data
- Can deal with noisy text
- Transparent model

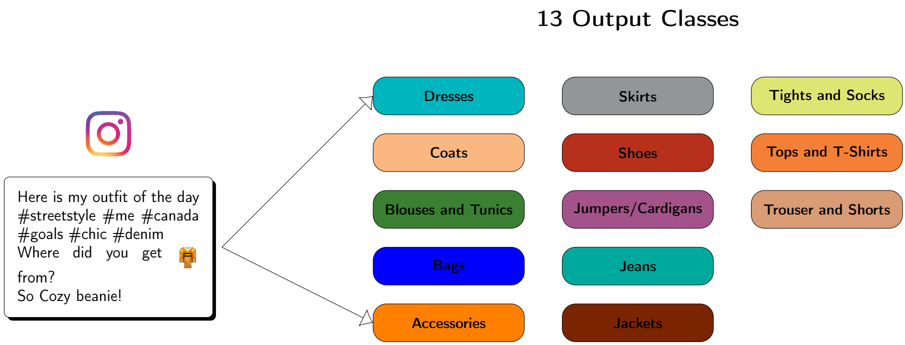
Cons:

- Require feature engineering
- Require a domain ontology

Can we reduce manual feature engineering and learn from data?

Problem: We don't have annotated data (yet)
Solution: Weakly supervised learning

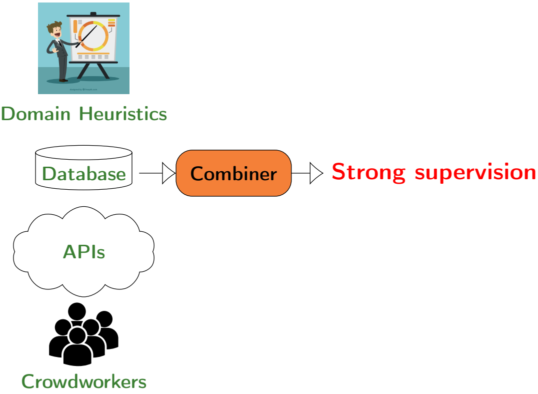
A multi-class multi-label classification problem



Alternative Sources of Supervision That Are Cheap but Weak

- Strong supervision: Manual annotation by expert
- Weak supervision: A signal that does not have full coverage/perfect accuracy

Sources of Weak Supervision



How To Combine Several Sources Of Weak Supervision?

- Simplest way to combine many weak signals: Majority Vote
- Recent research on combination of weak signals: Data Programming Paradigm[3]

Data Programming: Creating Large Training Sets, Quickly

Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, Christopher Ré
Stanford University
{ajratner,cdesa,senwu,dselsam,chrismre}@stanford.edu

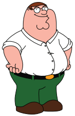
[3] Alexander J Ratner et al. "Data Programming: Creating Large Training Sets, Quickly". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., 2016, pp. 3567–3575. url: http://papers.nips.cc/paper/6523-data-programming-creating-large-training-sets-quickly.pdf.

Data Programming: Model Weak Supervision With Generative Model

- **Model weak supervision as labeling functions λ_i**
  - λ_i(unlabeled data) $\rightarrow$ label
- **Learn Generative Model $\pi_{\alpha,\beta}(\Lambda, Y)$ over the labeling process.**
  - Based on conflicts between labeling functions assign the functions an estimated accuracy α_i.
  - Based on empirical coverage of labeling functions assign the functions a coverage β_i.
- **Given α and β for each labeling function, it can be used to combine labels into a single probabilistic label**
  - Give more weight to high-accuracy functions
  - If there is a lot of disagreement$\rightarrow$ low probability label
  - If all labeling functions agree $\rightarrow$ high probability label

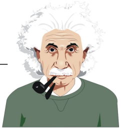
Data Programming Intuition

Low accuracy labeling functions **High accuracy labeling functions**



"it is not a coat"

"it is a coat"



Probabilistic Label: 0.6 probability that it is a coat

Majority Vote: 1.0 probability that it is not a coat

Pipeline for Weakly Supervised Classification in Instagram

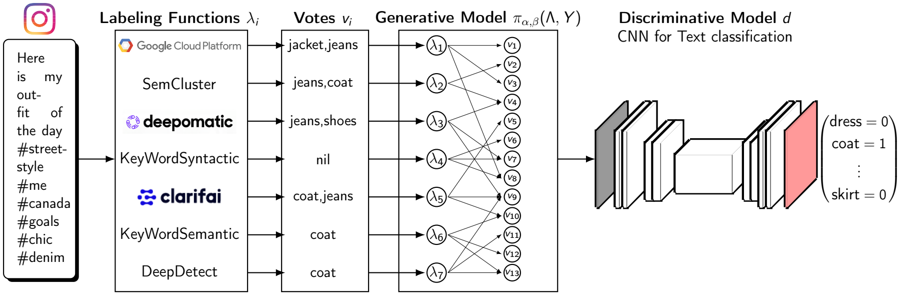


Figure: A pipeline for weakly supervised text classification of Instagram posts.

Extension of Data Programming to Multi-Label Classification

- **Problem: Data programming only defined for binary classification in original paper**
- **To make it work for multi-class setting:** model labeling function as $\lambda_i \to k_i \in \{0, \dots, N\}$ instead of $\lambda_i \to k_i \in \{-1, 0, 1\}$.
- **Idea 1 for multi-label**: model labeling function as $\lambda_i \to \vec{k_i} = \{v_0, \dots, v_n\} \wedge v_j \in \{-1, 0, 1\}$
- **Idea 2 for multi-label**: learn a separate generative model for each class, and let each labeling function give binary output for each class $\lambda_{i,j} \to k_{i,j} \in \{-1, 0, 1\}$.

Trained Generative Models: Labeling Functions' Accuracy Differ Between Classes

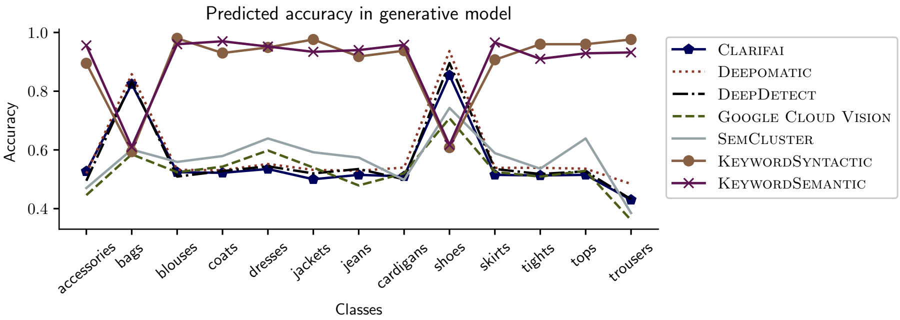


Figure: Multiple generative models can capture a different accuracy for labeling functions for different classes.

Discriminative CNN Model for Text Classification

- I have extended Kim Yoon's CNN model for text classification[4]
- To train the model with probabilistic labels produced by generative model, I use a *noise-aware* loss function[5]:

$$\frac{1}{N}\sum_{i=0}^{N} -(p(Y_i|\Lambda_i)\log(\sigma(\hat{y}_i)) + ((1 - p(Y_i|\Lambda_i))\log(1 - \sigma(\hat{y}_i)))) \quad (1)$$

[4] Yoon Kim. "Convolutional Neural Networks for Sentence Classification". In: *EMNLP*. ACL, 2014, pp. 1746–1751.

[5] N is the number of classes, $p(Y_i|\Lambda_i)$ is the probabilistic labels for class i, and $\hat{y}_i$ is the logits for class i

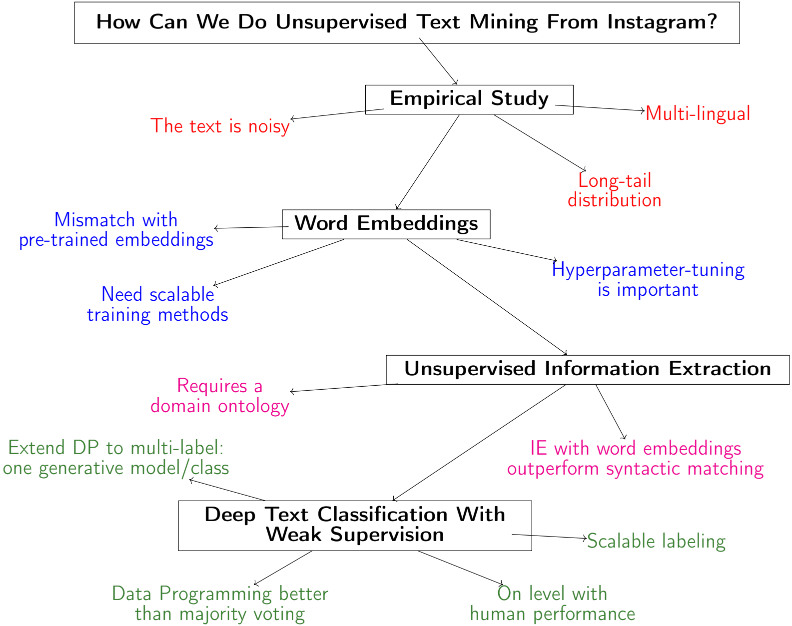
Data Programming Beats Majority Voting and the Multi-Channel Model Was Not Useful

- **Data programming gives 6 F_1 points improvement over majority vote[6], achieving an F_1 score of 0.61 (On level with human performance)**

Model	*Accuracy*	*Precision*	*Recall*	*Micro-F_1*	*Macro-F_1*	*Hamming Loss*
CNN-DataProgramming	**0.797** $\pm$ 0.01	**0.566** $\pm$ 0.05	0.678 $\pm$ 0.04	**0.616** $\pm$ 0.02	**0.535** $\pm$ 0.01	**0.195** $\pm$ 0.02
CNN-MajorityVote	0.739 $\pm$ 0.02	0.470 $\pm$ 0.06	**0.686** $\pm$ 0.05	0.555 $\pm$ 0.03	0.465 $\pm$ 0.05	0.261 $\pm$ 0.03

- **Main cause of error: data sparsity (can not extract clothing items from the text if it is never mentioned in the text)**

[6] A smaller, hand-labeled dataset by experts was used for evaluation


How Can We Do Unsupervised Text Mining From Instagram?
Empirical Study
The text is noisy
Multi-lingual
Long-tail distribution
Word Embeddings
Mismatch with pre-trained embeddings
Need scalable training methods
Hyperparameter-tuning is important
Unsupervised Information Extraction
Requires a domain ontology
IE with word embeddings outperform syntactic matching
Extend DP to multi-label: one generative model/class
Deep Text Classification With Weak Supervision
Scalable labeling
Data Programming better than majority voting
On level with human performance

Conclusion

- Instagram text is just as noisy as Twitter, comment sections are multi-lingual, long tail text distribution
- Word Embeddings are useful for IE, especially in social media
- Deep learning with weak supervision and data programming is a promising approach for text mining in social media

Main line of future work:

Combine text analytics with image analysis[7]

Thanks:

Shatha Jaradat, Nima Dokoohaki Ph.D, Prof. Mihhail Matskin

[7] Shatha Jaradat. "Deep Cross-Domain Fashion Recommendation". In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. RecSys '17. Como, Italy: ACM, 2017, pp. 407–410. isbn: 978-1-4503-4652-8. doi: 10.1145/3109859.3109861. url: http://doi.acm.org/10.1145/3109859.3109861.

Questions