ROBUST LANGUAGE LEARNING VIA EFFICIENT BUDGETED ONLINE ALGORITHMS

Simone Filice, **Giuseppe Castellucci**, Danilo Croce, Roberto Basili
University of Roma, Tor Vergata
Roma – Italy

SENTIRE 2013 (ICDM 2013)
Dallas, December 7

Motivations

- In the Web people generate **continuously** new contents
  - even more in the Social Media era
- **Adapting** models to dynamic environments
  - concept shifting

- Contents are mainly expressed in natural language
- **Kernel** methods
  - provide robust methods to analyze NL
  - but slow performances are experimented

- **Sentiment analysis** over Twitter
  - tracking user moods
  - dynamic context

Outline

- Online Learning
  - Passive Aggressive
- Kernel methods & OL
- Budgeted Learning
- Fairness and Weight Adjustment
- Sentiment Analysis over Twitter
  - Feature modeling
  - Distributional features
  - Results
- Conclusions

Online Learning

- A ML model that uses a new unique example at each iteration.
- Open-ended vision
  - trying to adapt to an environment
- Obviously, trying to maintain good performance levels.

Why?

- Initial dataset not enough representative
  - data distribution change over time
- Building from zero a new model not feasible
  - batch procedure high computational costs

Passive-Aggressive [Crammel et al '06]

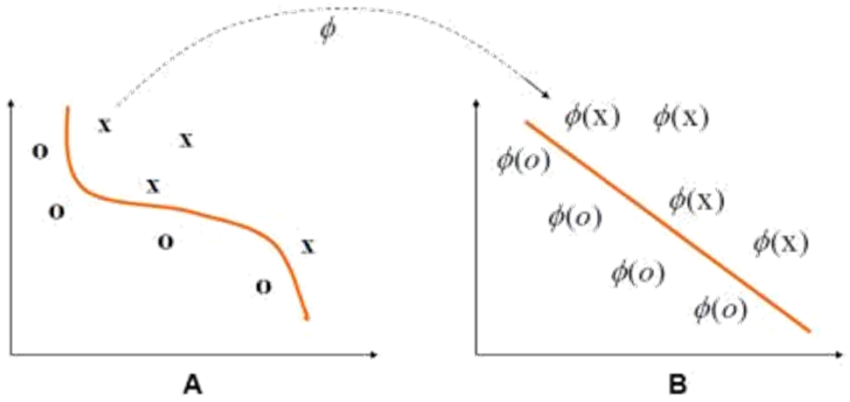
- Every time a new example is available the current classification function is modified

Classification function
$f_t(\boldsymbol{x}) = \boldsymbol{w}_t{}^T \boldsymbol{x}$
Optimization Problem (PA-I)
$\boldsymbol{w}_{t+1} = argmin_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}_t\|^2 + C\xi$ Such that $1 - y_t \mathrm{f}_t(\boldsymbol{x}_t) \leq \xi, \xi \geq 0$
Close form solution
$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \tau_t y_t \boldsymbol{x}_t$ where $\tau_t = \min\left\{C; \frac{\max(0, 1 - y_t f_t(\boldsymbol{x}_t))}{\|\boldsymbol{x}_t\|^2}\right\}$

- Modify the model **as less as possible**
- The model does not need to be re-generated from scratch
- Capability of tracking a **Shifting Concept**

Kernel methods

- Implicit mapping in a **richer** feature space defined by the kernel function



- Enable
  - Non linear classification functions
  - Structured representations
- Widely used in NLP [Collins and Duffy, 2001; Cristianini et al., 2002; Croce et Al, 2011]

Kernel methods and OL

- Passive-Aggressive with Kernel
- Add support vector to the model when needed

Classification function
$f_t(x) = \sum_{i \in S} \alpha_i k(x, x_i)$
Optimization Problem (PA-I)
$f_{t+1}(x) = \text{argmin}_f \frac{1}{2} \|f(x) - f_t(x)\|^2_{\mathcal{H}} + C\xi$ Such that $1 - y_t f_t(x_t) \leq \xi, \xi \geq 0$
Close form solution
$f_{t+1}(x) = f_t(x) + \alpha_t k(x, x_t)$ where $\alpha_t = y_t \cdot \min\left\{C; \frac{\max(0, 1 - y_t f_t(x_t))}{\|x_t\|^2_{\mathcal{H}}}\right\}$

- Drawbacks
  - **Memory consumption:** all the support vectors must be stored
  - **Classification costs**
- Unacceptable in a continuous learning environment

Budgeted Learning

- One possible solution
  - limit the maximum number of Support Vectors
  - discarding old SV when the budget is exceeded

Passive Aggressive on a Budget [Wang and Vucetic 2010]:

When the budget is saturated the objective function is slighly modified, adding a new constraint:

$$f_{t+1}(x) = \text{argmin}_f \frac{1}{2}\|f(x) - f_t(x)\|^2_{\mathcal{H}} + C\xi$$

Such that: $1 - y_t f_t(x_t) \leq \xi,\ \xi \geq 0$

$$f(x) = f_t(x) - \underbrace{\alpha_r k(x_r, x)}_{SV\ elimination} + \underbrace{\sum_{i \in V} \beta_i k(x_i, x)}_{weights\ modification}$$

Where r is the index of the support vector to remove and $V \subseteq S \cup \{t\} \setminus \{r\}$ is the set of the indices of the support vectors whose weights can be modified

Budgeted Learning cont'd

- Different BPA strategies
  - BPA-Simple: $V = \{t\}$
  - BPA-Projecting: $V = SV_t \cup \{t\} - \{r\}$
  - BPA-Nearest-Neighbor: $V = \{t\} \cup NN(r)$, where $NN(r)$ is the index of the nearest neighbor of x_r
- In this work:
  - BPA-NN using the kernel similarity measure to calculate NN, instead of Euclidean distance

Fairness & Weight Adjustment

- **Fairness**
  - balance positive and negative contributions during training

- Modify the objective function

$$f_{t+1}(x) = \text{argmin}_f \frac{1}{2}\|f(x) - f_t(x)\|^2_{\mathcal{H}} + C(y_t)\xi$$

$$\text{Such that } \; 1 - y_t f_t(x_t) \leq \xi, \; \xi \geq 0$$

$$\text{Where } C(y_t) = \begin{cases} C_+ \; if \; y_t = +1 \\ C_- \; if \; y_t = -1 \end{cases}$$

  - C_+ and C_- are choosen so that, the potential cost of false positives equals the potential cost of false negatives:
    - $\frac{C_+}{C_-} = \frac{N(negative)}{N(positive)}$

Fairness & Weight Adjustment

- **Weight Adjustment**
  - mitigate the effect of changing the model at each misclassification
    - as it can produce instability in the model (e.g. outliers)
  - Multiple iterations on the whole training set
    - OL will lose its computational advantage (w.r.t.batch learners)

  - WA: variation of multiple iterations
    - Only the SVs are exploited again
    - SVs set doesn’t change, only the weights of each SV

Sentiment Analysis over Twitter

- Twitter has the numbers
  - More than 100 million active users worldwide
  - Highly **dynamic** place: more than 5K tweets each second
- In Twitter people share
  - news
  - ideas
  - opinions
- An emerging mine of information for
  - companies
  - politicians
  - other users

Modeling Sentiment

- In Natural Language Learning we aim at capturing the similarity between examples
  - redundantly capture different levels of lexical similarity between tweets
- Exploit kernel combination
  - Each **kernel** works over a feature representation
  - Each **representation** aims at capturing different semantic aspects
- Here, two representation:
  - Bag-of-Word features
  - Distributional features

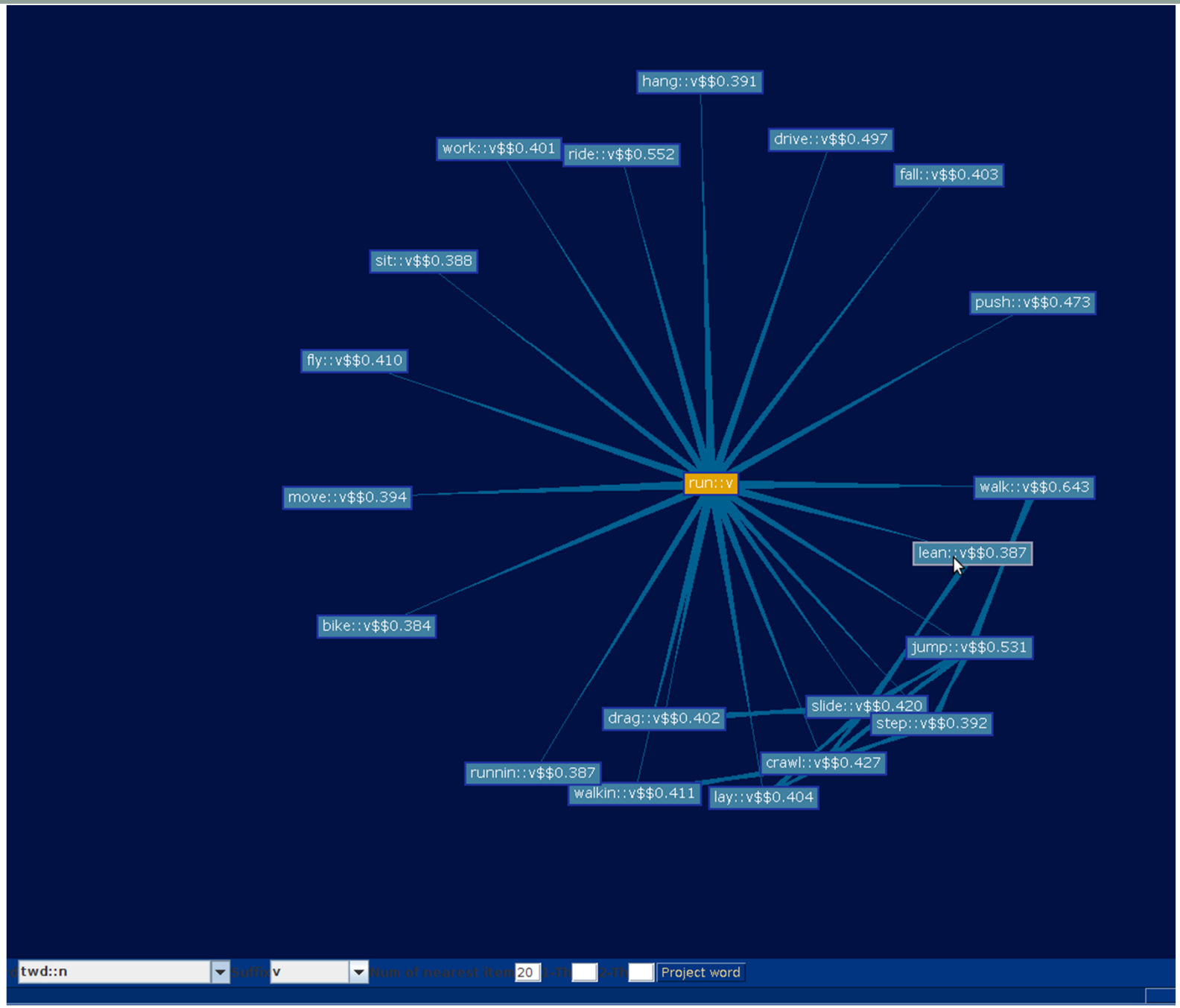
Bag-of-Word

- Emphasize pure lexical information (BOW)
  - word overlap between tweets
  - boolean weighting schemas on a vector
  - each dimension represents a ⟨lemma, POS⟩ pair

- Lemma and Part-Of-Speech from Natural Language Processor (Basili and Zanzotto, 2002) customized for tweets:
  - Normalizer
  - Tokenizer
  - Morphology Analyzer
  - Part of Speech Tagger

Distributional features

- Generalize lexical information
  - Reducing data sparseness
- Unsupervised analysis
  - More than 3 million of tweets
- Contruction of a Word Space (Sahlgren, 2006)
  - Word-by-Context matrix
  - Contexts 3 words around a target word
- Latent Semantic Analysis (LSA):
  - $M = USV^T$
  - Approximation considering the first k=250 components
- A word is projected in this reduced Word Space
- A tweet is represented as the «sum» of its words (Cristianini et al, 2002)


hang::v$$0.391
work::v$$0.401
ride::v$$0.552
drive::v$$0.497
fall::v$$0.403
sit::v$$0.388
push::v$$0.473
fly::v$$0.410
run::v
move::v$$0.394
walk::v$$0.643
lean::v$$0.387
bike::v$$0.384
jump::v$$0.531
slide::v$$0.420
drag::v$$0.402
step::v$$0.392
crawl::v$$0.427
runnin::v$$0.387
walkin::v$$0.411
lay::v$$0.404
twd::n
Suffix
v
Num of nearest item
20
Project word

Kernel functions

- Linear kernel (lin)
  - Cosine similarity: $\sigma(w_1, w_2) = \frac{\overrightarrow{w_1} * \overrightarrow{w_2}}{\|\overrightarrow{w_1}\| \, \|\overrightarrow{w_2}\|}$
  - Applied to BOW and LSA representations
- Radial Basis Function (rbf) kernel
  - $RBF(w_1, w_2) = e^{-\gamma \|\overrightarrow{w_1} - \overrightarrow{w_2}\|^2}$
  - Applied to LSA representation

- Finally, **kernel combination**
  - Linearly combine kernel contribution
    - E.g. $\alpha * lin(bow(w_1), bow(w_2)) + \beta * rbf(lsa(w_1), lsa(w_2))$

Dataset and learning parameters

- SemEval 2013 Task 2 corpus is used
  - Message polarity classification task
- Classify a tweet w.r.t. *positive*, *negative* and *neutral* classes
- 10205 training tweets and 3813 testing tweets

- Parameters
  - Repeated Random Sub-sampling Validation
  - 10-fold cross-validation strategy on a subset of training data according to a 70%-30% split

- In OL, **data order matters**
  - 10 different models are computed on different shuffled version of data
  - measure used in SemEval: mean between positive and negative F1

Results BPA

Budget	BoW(lin)	LSA(lin)	BoW(lin)+LSA(lin)	BoW(lin)+LSA(rbf)	Saving w.r.t. SVM
100	0,396±0,041	0,451±0,041	0,444±0,080	0,430±0,063	98%
250	0,408±0,029	0,469±0,029	0,462±0,056	0,444±0,055	95%
500	0,443±0,021	0,480±0,021	0,513±0,029	0,516±0,044	89%
750	0,448±0,030	0,495±0,030	0,500±0,044	0,503±0,057	84%
1000	0,446±0,023	0,488±0,023	0,517±0,030	0,525±0,034	80%
1500	0,448±0,037	0,496±0,037	0,516±0,038	0,506±0,047	71%
2000	0,463±0,034	0,494±0,034	0,530±0,043	0,520±0,045	62%
2500	0,478±0,034	0,501±0,031	0,525±0,044	0,532±0,043	55%
3000	0,484±0,023	0,496±0,023	0,548±0,053	0,544±0,054	49%
3500	0,496±0,008	0,521±0,008	0,546±0,050	0,568±0,043	43%
4000	0,496±0,027	0,513±0,027	0,559±0,046	0,572±0,038	38%
4500	0,504±0,013	0,538±0,013	0,564±0,048	0,574±0,048	35%
5000	0,505±0,018	0,535±0,018	0,572±0,043	0,579±0,040	32%
6000	0,505±0,021	0,543±0,021	0,581±0,037	0,593±0,036	28%
7000	0,510±0,020	0,548±0,020	0,580±0,041	0,595±0,037	28%
8000	0,516±0,017	0,550±0,017	0,580±0,041	0,597±0,035	28%
9000	0,516±0,017	0,550±0,017	0,580±0,041	0,597±0,035	28%
10000	0,516±0,017	0,550±0,017	0,580±0,041	0,597±0,035	28%

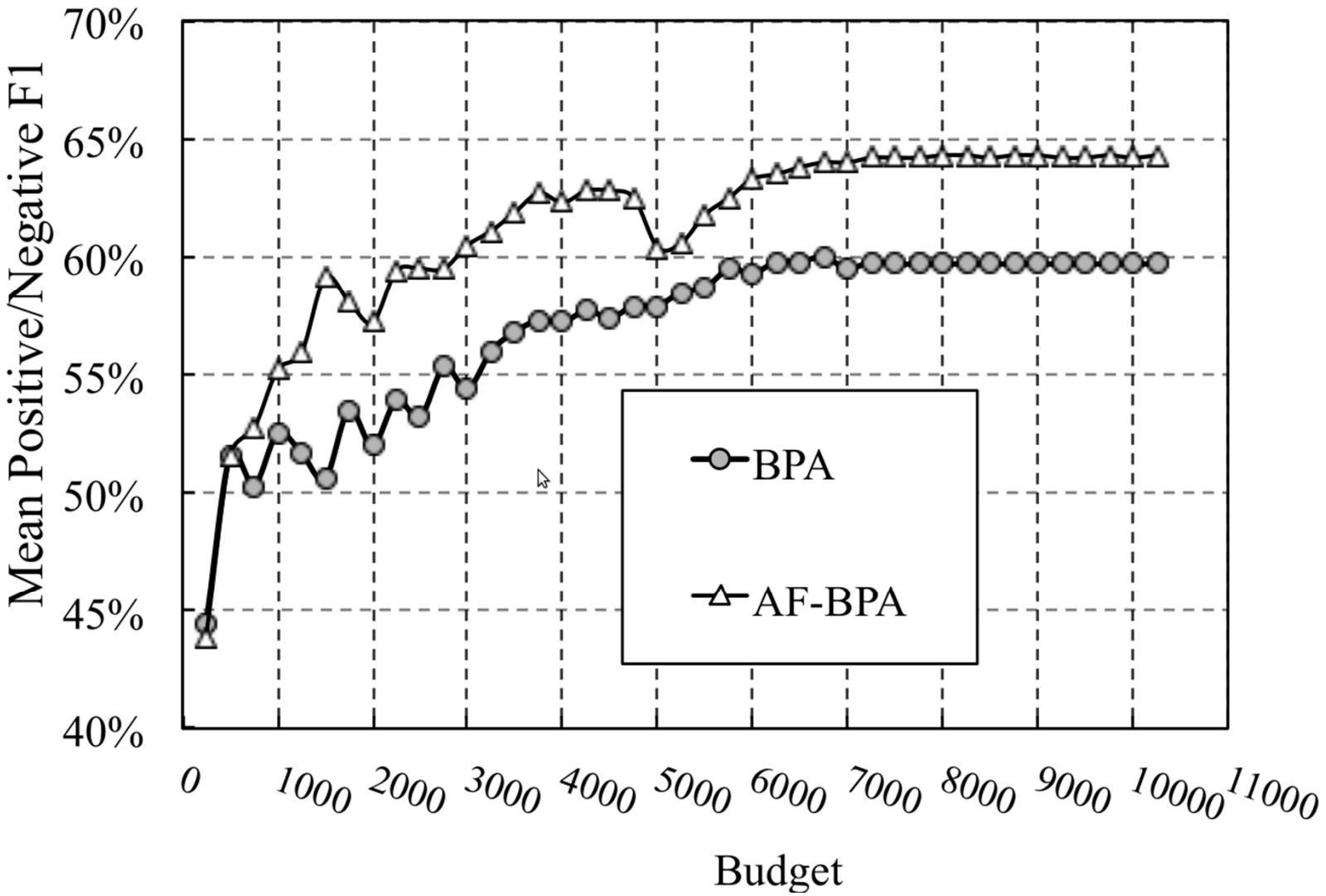
SVM	Best system @ Semeval2013
0.6541	0.6902

Results AF-BPA

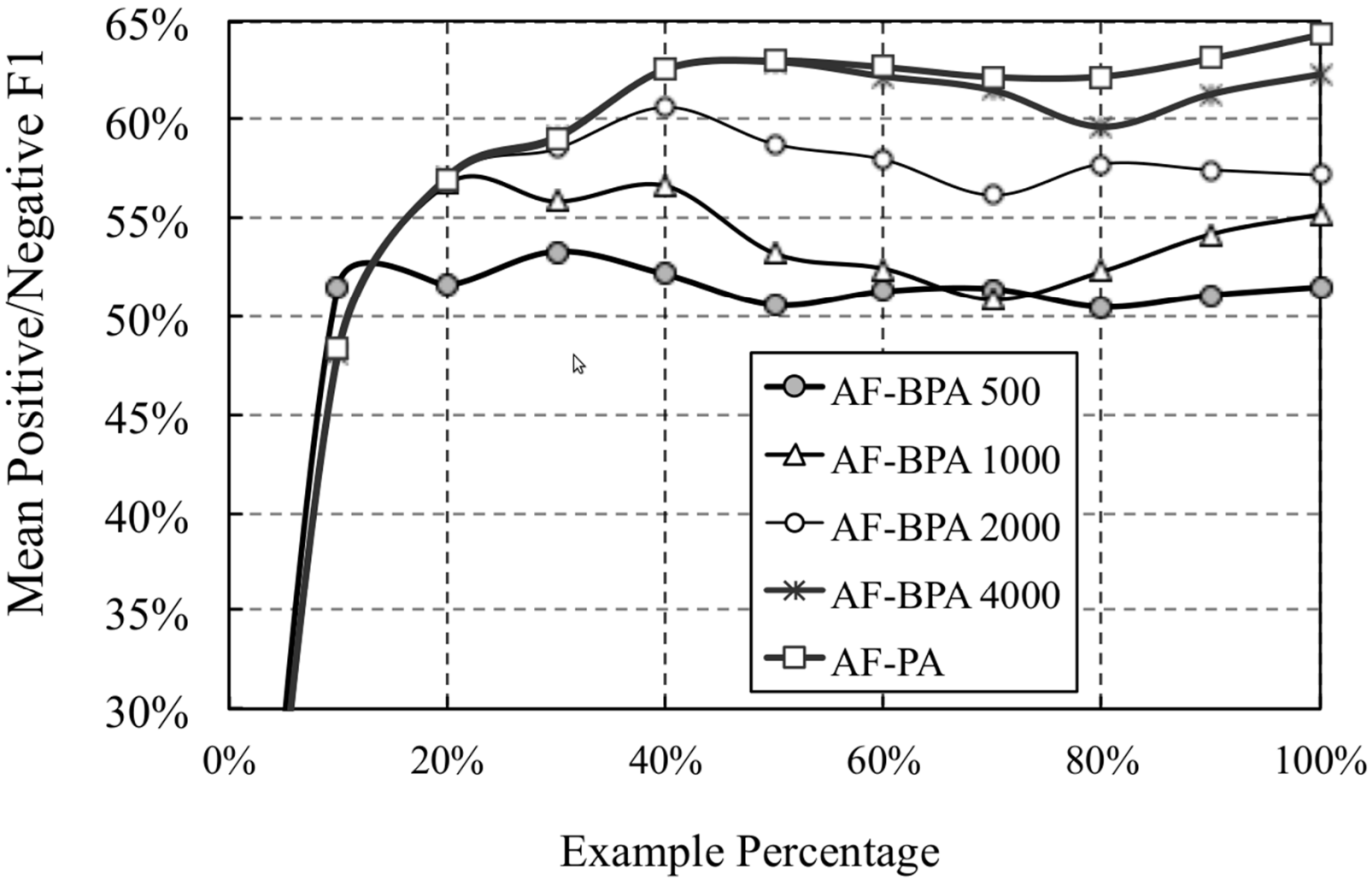
Budget	BoW(lin)	LSA(lin)	BoW(lin)+LSA(lin)	BoW(lin)+LSA(rbf)	Saving w.r.t. SVM
100	0,394±0,030	0,346±0,073	0,456±0,055	0,471±0,035	98%
250	0,418±0,029	0,454±0,059	0,505±0,023	0,438±0,060	95%
500	0,412±0,041	0,459±0,060	0,503±0,049	0,516±0,038	89%
750	0,426±0,039	0,451±0,055	0,542±0,037	0,527±0,031	84%
1000	0,475±0,022	0,511±0,040	0,542±0,034	0,552±0,025	80%
1500	0,471±0,024	0,517±0,035	0,555±0,025	0,591±0,029	70%
2000	0,490±0,030	0,564±0,020	0,585±0,019	0,572±0,024	62%
2500	0,498±0,026	0,585±0,008	0,599±0,011	0,595±0,025	54%
3000	0,541±0,017	0,582±0,011	0,600±0,016	0,605±0,029	47%
3500	0,548±0,012	0,574±0,012	0,609±0,012	0,619±0,012	41%
4000	0,556±0,019	0,573±0,014	0,621±0,009	0,623±0,012	35%
4500	0,569±0,018	0,573±0,019	0,618±0,020	0,628±0,009	30%
5000	0,576±0,008	0,574±0,013	0,610±0,015	0,604±0,015	24%
6000	0,545±0,013	0,573±0,013	0,626±0,007	0,633±0,008	18%
7000	0,576±0,005	0,572±0,005	0,634±0,006	0,640±0,004	14%
8000	0,575±0,004	0,574±0,006	0,635±0,004	0,643±0,004	14%
9000	0,578±0,004	0,574±0,008	0,636±0,005	0,643±0,005	14%
10000	0,577±0,004	0,575±0,007	0,636±0,005	0,642±0,004	14%

SVM	Best system @ Semeval2013
0.6541	0.6902

Results


70%
65%
60%
55%
50%
45%
40%
Mean Positive/Negative F1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
Budget
BPA
AF-BPA

Results


Mean Positive/Negative F1
65%
60%
55%
50%
45%
40%
35%
30%
0%
20%
40%
60%
80%
100%
Example Percentage
AF-BPA 500
AF-BPA 1000
AF-BPA 2000
AF-BPA 4000
AF-PA

Conclusions

- Semantic kernel functions in OL schema
- Fairness and Weight adjustment
  - Our best performance in 4° position in SemEval2013 Task 2 rank
- Fast deployment of systems as
  - **no particular artificial features** manually encoded
  - **no manually coded resource** (e.g. no sentiment lexicon)

- In the future
  - verify the applicability of more complex kernel functions
  - compute automatically parameters of kernel combination

Thank you, questions?