Sentiment polarity classification using statistical data compression models

Dominique Ziegelmayer Rainer Schrader

University of Cologne
Institute of Computer Science
Weyertal 80
50931 Köln

December 10, 2012

Classification using compression models

Informal idea

- Compression algorithms build up extensive statistics
- Homogeneous data leads to better compression ratios
- Given categories $C_1, \ldots, C_n$ and corresponding training sets $T_1, \ldots, T_n$, affiliation of some document d can be determined by analyzing joint-compression ratio of each T_i with d

The information theoretic view

Cross entropy

- Measure for similarity of two sources (probability distributions)
- Gives average number of bits (per symbol) to identify an event using a probability distribution Q, rather than true distribution P
- Classification evaluates cross entropy between P for the source of document d and Q given by the compression model
- Exact value hard to compute $\rightarrow$ In practice mostly estimated

PPM (Cleary and Witten, 1984)

- Used in popular implementations such as RAR or 7Zip
- Remains among best compression algorithms for natural text

Basic concept

- Predict a symbol x_i by context $c_{i,j} = \{x_{i-j}, x_{i-j+1}, \ldots, x_{i-1}\}$ of order j using probability distribution p_j
- If $(c_{i,j}, x_i)$ not present in model of order j, add to model, update p_j, switch to order $j-1$ and encode order switch in output
- Else update p_j, encode x_i, reset order to j and restart with x_{i+1}

C-Measure (Hunnisett and Teahan, 2004)

- Based on the PPM compression algorithm but uses fixed order j
- Slightly outperforms PPM on the topic classification task

Basic Concept

- Extract all strings $c_{i,j} \circ x_i$ (features) from document d
- Add 1 to result if string is present in the training set T_i
- Assign d to class C_i with highest score

C_k-Measure (Ziegelmayer and Schrader, 2012)

- Keeps computational properties of C-measure
- Optimization for binary sentiment classification
- Omit features occurring in both classes (with similar frequencies)

Basic Concept

- Extraction and classification analogous to C-measure but:
- Add 1 to result if string is k-times more frequent in T_+ than in T_-

F_k-measure (Ziegelmayer and Schrader, 2012)

- Keeps computational properties of C-measure and (implicit) feature selection of C_k-measure
- Counts frequency of features rather than existence

Basic Concept

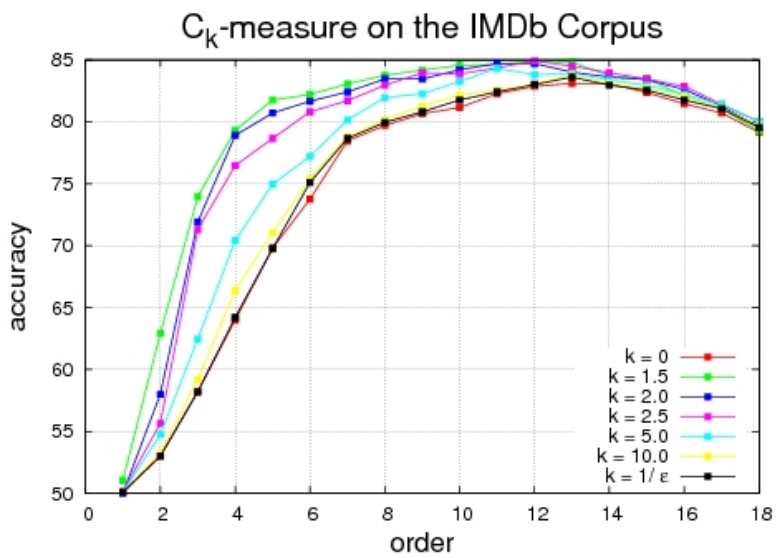
- Works analogous to C_k-measure but for each string $c_{i,j} \circ x_i$:
- Add absolute frequency of $c_{i,j} \circ x_i$ in corresponding model (T_-, T_+) to result instead of 1 for pure existence

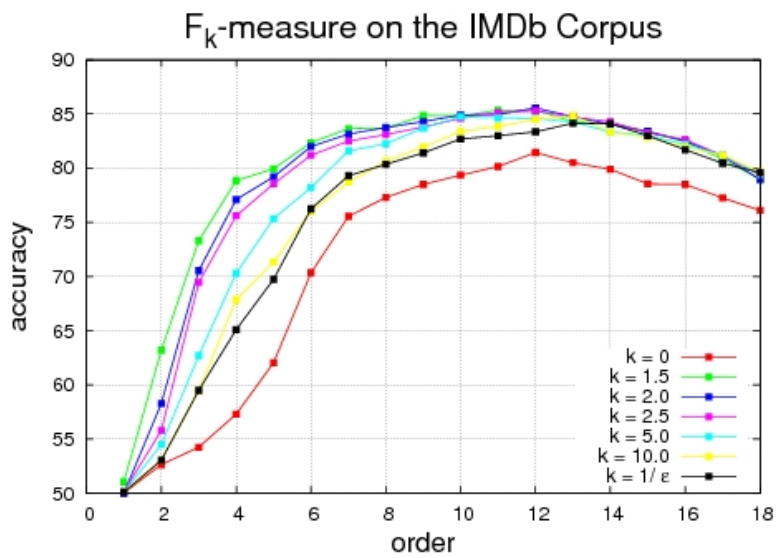
Corpora employed

- IMDb corpus (`polarity dataset v2.0` by Pang and Lee)
  - Large corpus (2,000 documents, 7,786,004 characters) with a rather versatile and complex language
- Amazon corpus (Custom dataset created from amazon.com)
  - Mid-Size corpus (2,000 documents, 682,124 characters) with mostly homogeneous and less complex language
- Twitter corpus (Public dataset from Sanders Analytics)
  - Small corpus (1,000 documents, 97,261 characters) with informal and rather simple language

Results on the IMDb corpus

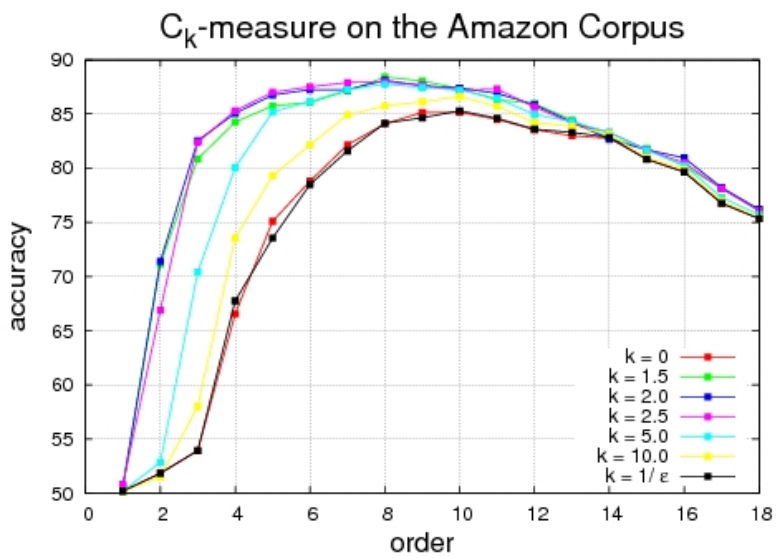
No	Method	Accuracy
(1)	PPMd	82.35%
(2)	C_0-measure	83.10%
(3)	$C_{2.5}$-measure	84.90%
(4)	$F_{2.5}$-measure	85.30%
(5)	**SVM (pres. unigram)**	**86.35%**

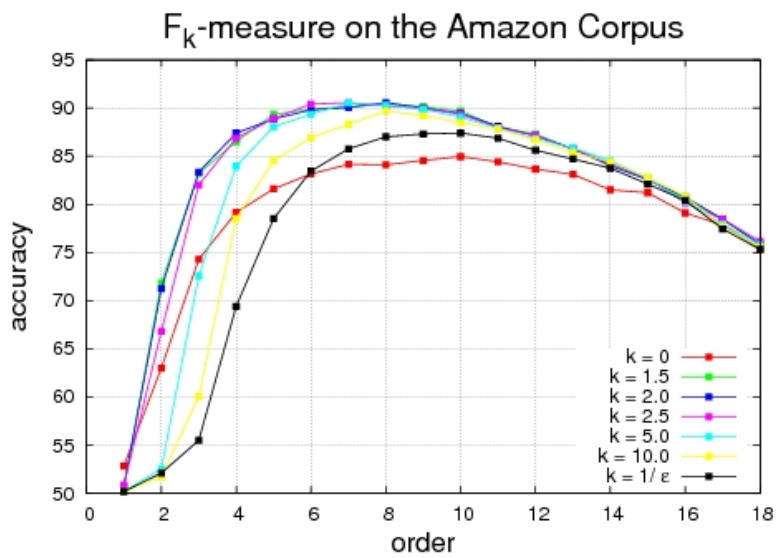




Results on the Amazon corpus

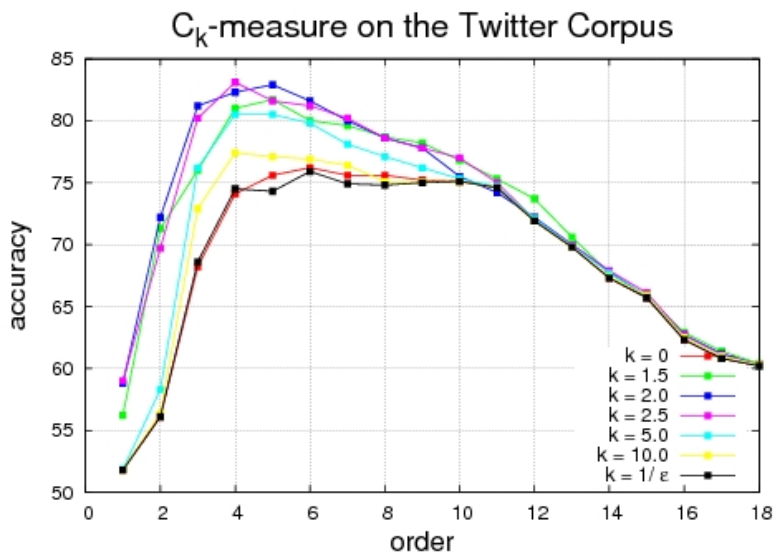
No	Method	Accuracy
(1)	PPMd	86.15%
(2)	C_0-measure	85.15%
(3)	$C_{2.5}$-measure	87.95%
(4)	**$F_{2.5}$-measure**	**90.55%**
(5)	SVM (pres. unigram)	86.35%

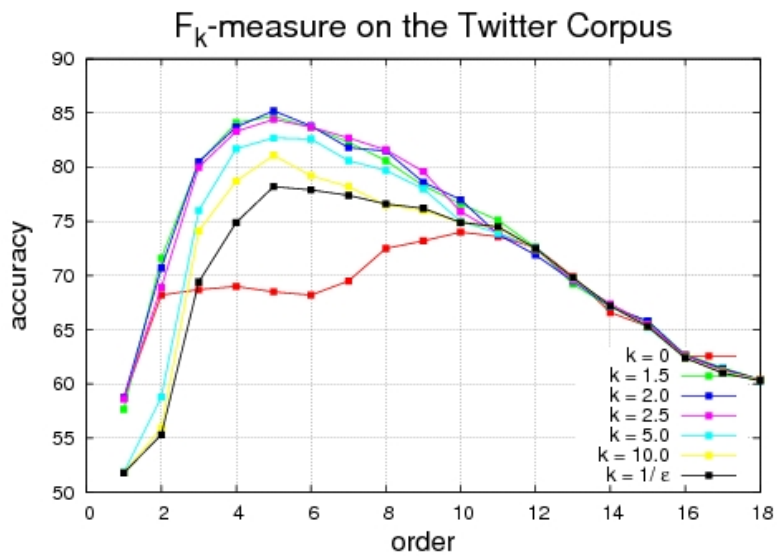




Results on the Twitter corpus

No	Method	Accuracy
(1)	PPMd	78.80%
(2)	C_0-measure	76.20%
(3)	$C_{2.5}$-measure	83.10%
(4)	**$F_{2.5}$-measure**	**84.40%**
(5)	SVM (pres. unigram)	77.80%





Discussion (1)

Why compression based sentiment classification?

- Requires no preprocessing and is easy to apply
- k-measures efficient in time and space complexity
- $F_{2.5}$-measure achieved 90.55% on Amazon corpus and outperformed SVM on the Twitter corpus by more than 6%

Discussion (2)

Why are *k*-measures performing better?

- We found misclassifications and spelling mistakes especially in Amazon and Twitter corpus

➨ *k*-measures effectively eliminate noise in the model

➨ Cope better with spelling mistakes and informal language

Discussion (3)

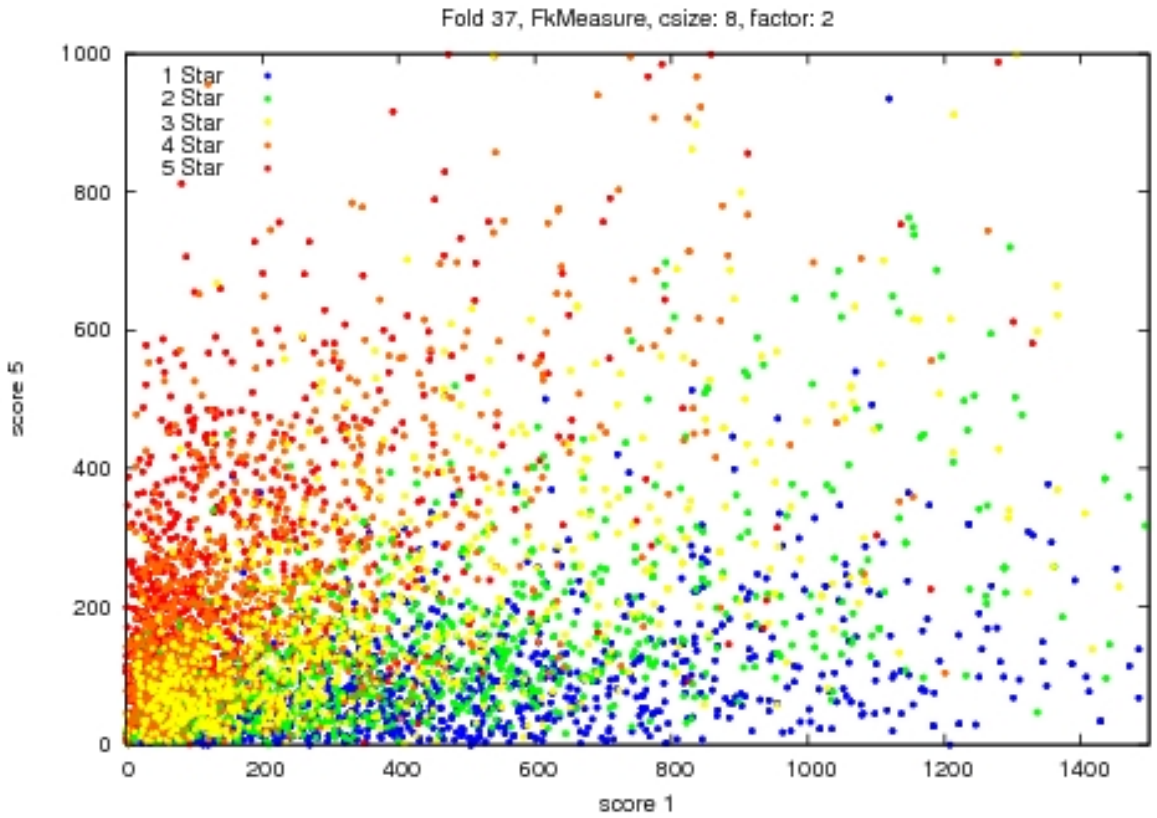
Interesting findings for future work

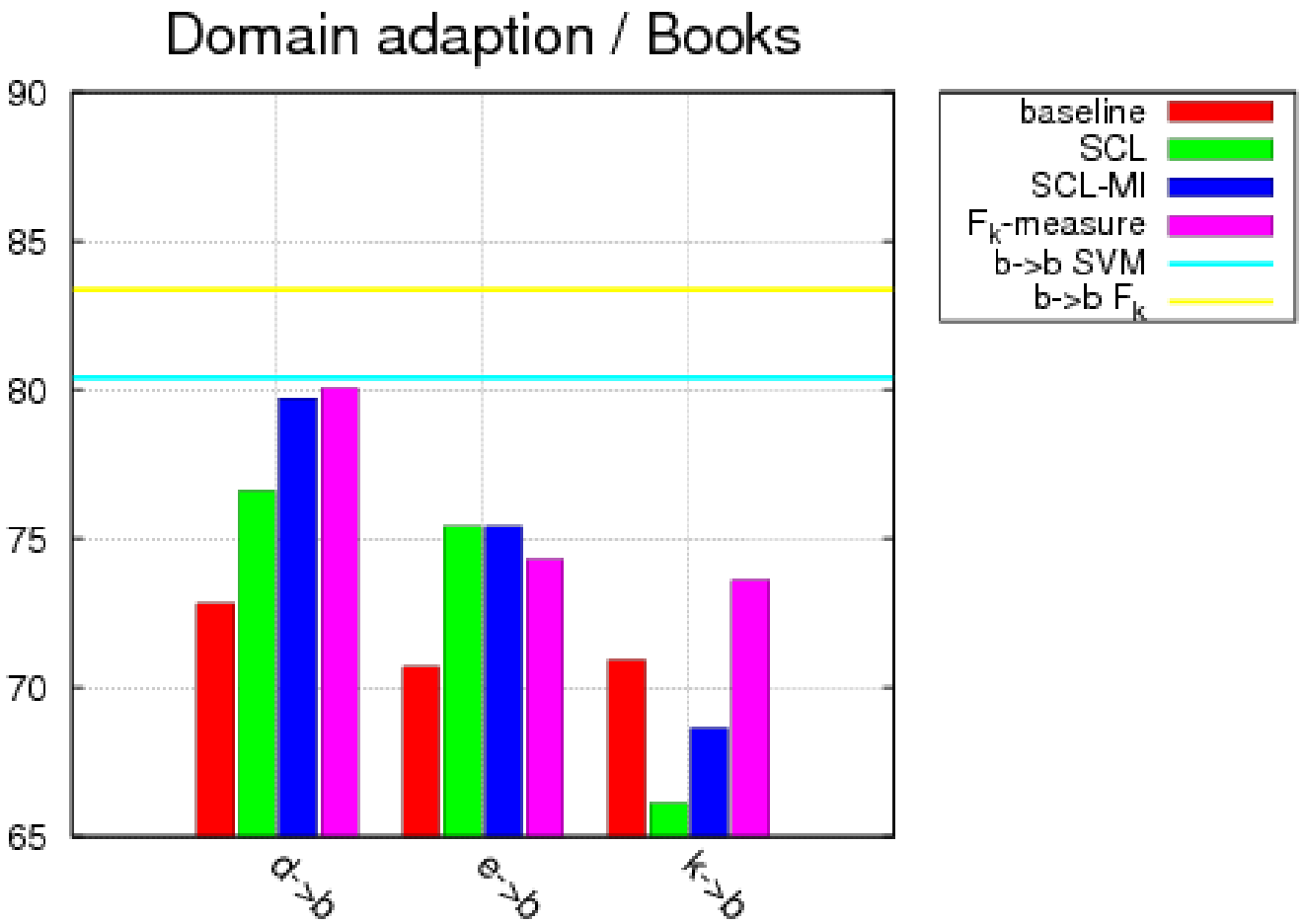
- Regression seems possible using ratio between positive and negative scores
- Cross-domain polarity classification performance seems to be slightly better than standard approach
- Character based approaches seem to obtain better results in inflective languages (McNamee et al.)

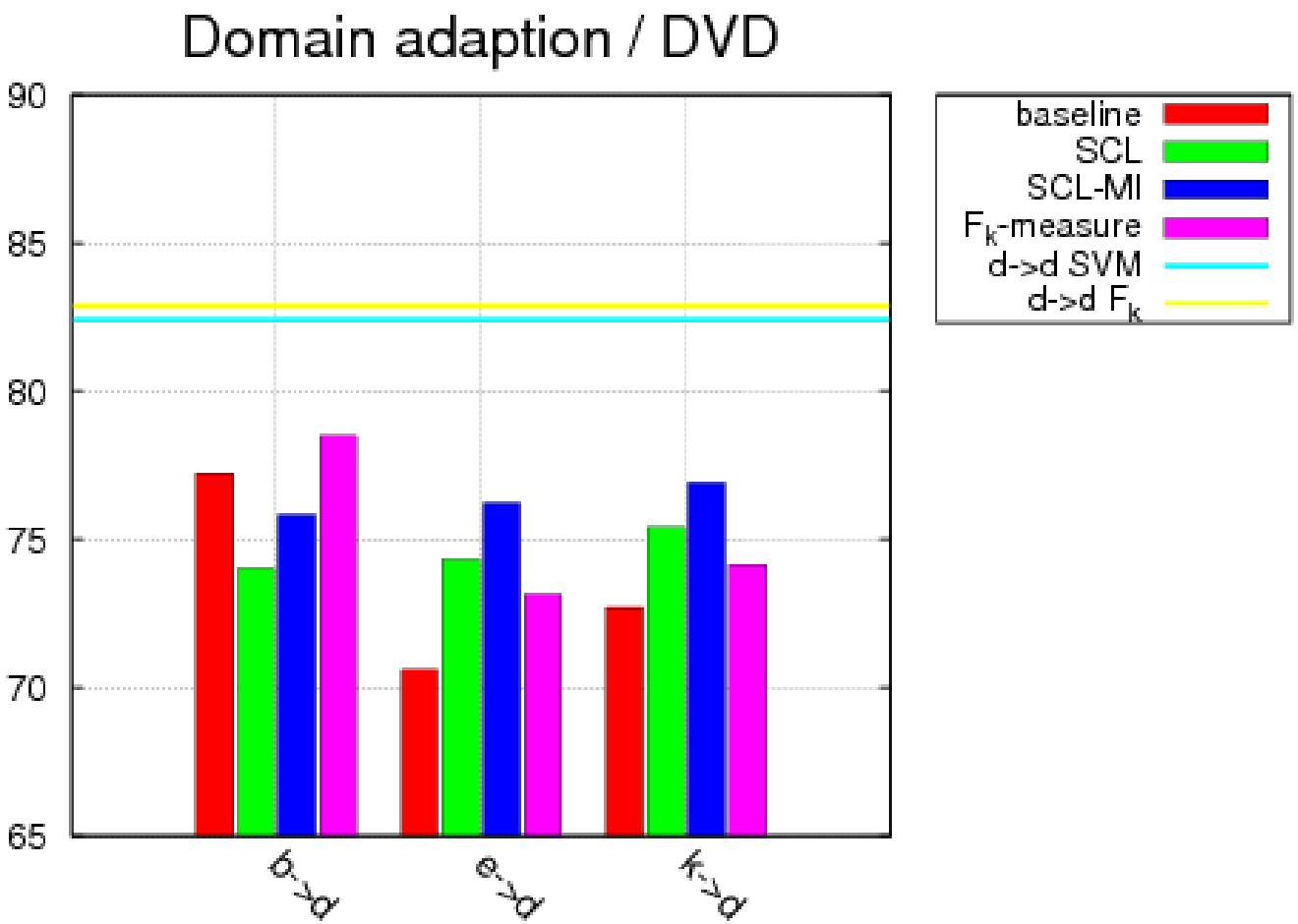
Backup

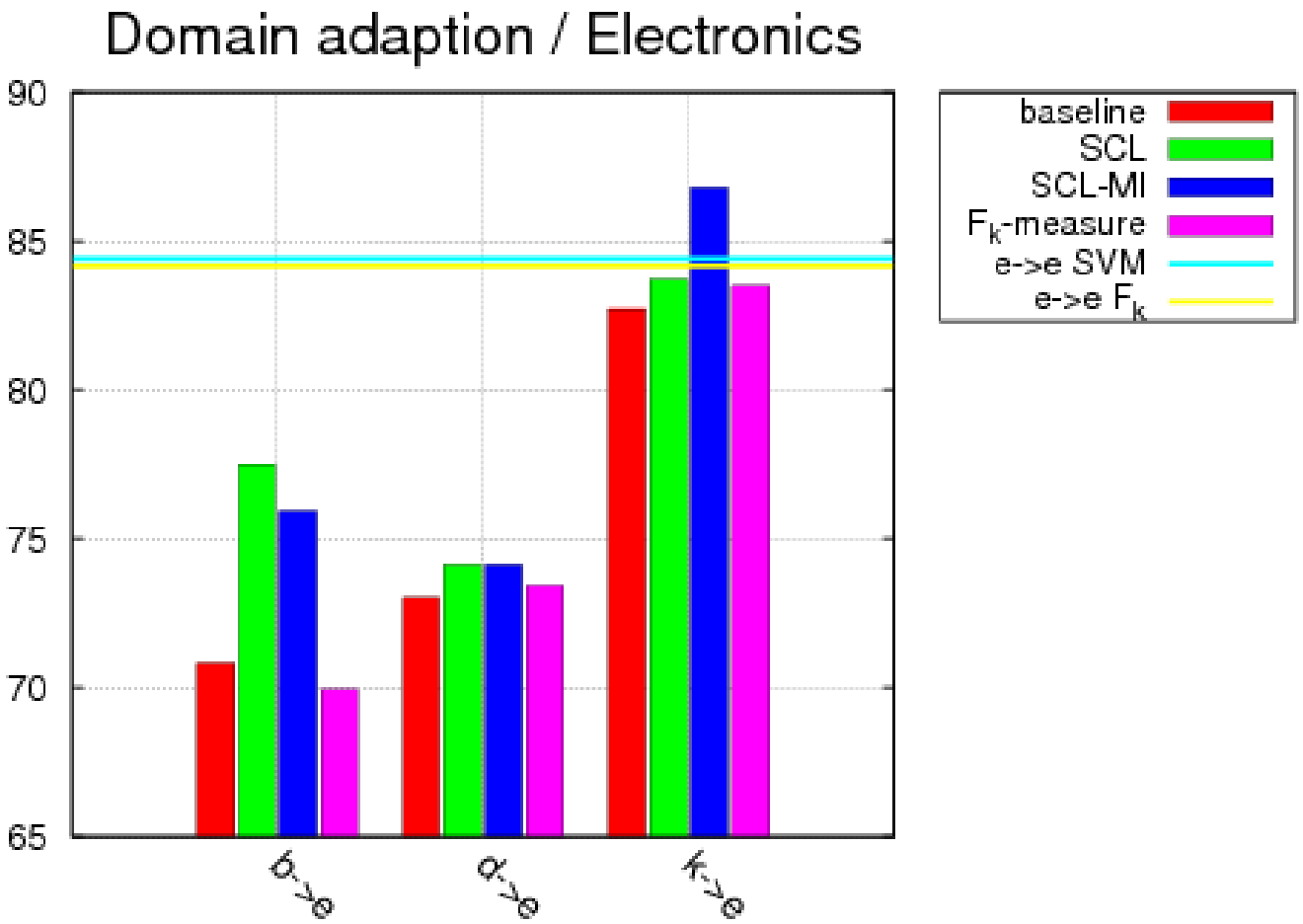
Regression using F_k-measure

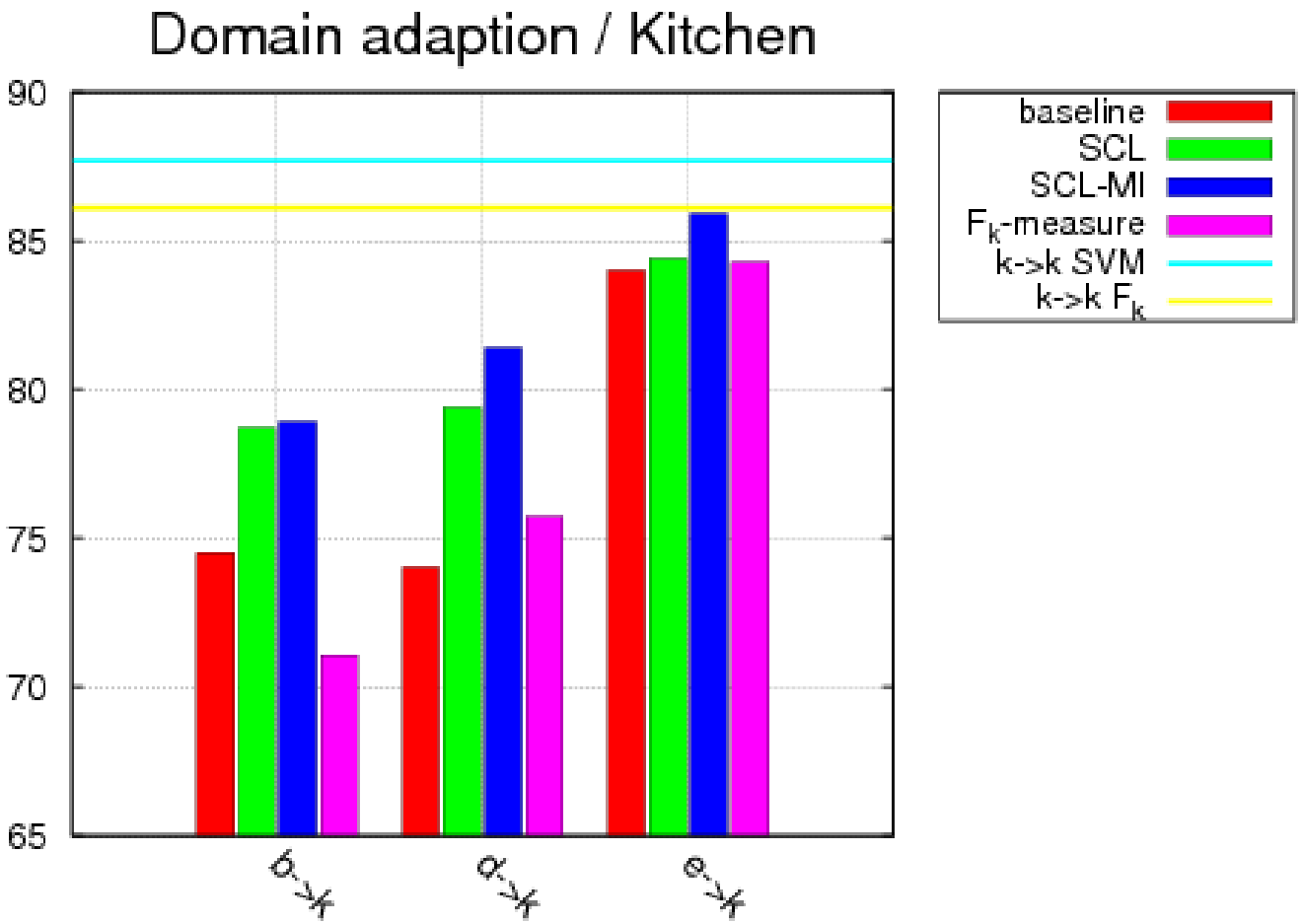
- Trained with 1-Star and 5-Star only, Tested with all reviews
- Star-rating shows linear order (1 < 2 < 3 < 4 < 5)




Domain adaption / Books
90
85
80
75
70
65
d->b
e->b
k->b
baseline
SCL
SCL-MI
F_k-measure
b->b SVM
b->b F_k


Domain adaption / DVD
90
85
80
75
70
65
b->d
e->d
k->d
baseline
SCL
SCL-MI
F_k-measure
d->d SVM
d->d F_k


Domain adaption / Electronics
90
85
80
75
70
65
b->e
d->e
k->e
baseline
SCL
SCL-MI
F_k-measure
e->e SVM
e->e F_k


Domain adaption / Kitchen
90
85
80
75
70
65
b->k
d->k
e->k
baseline
SCL
SCL-MI
F_k-measure
k->k SVM
k->k F_k

IMDb corpus

IMDb corpus (`polarity dataset v2.0` by Pang and Lee)

- 2,000 reviews written by 312 authors
- Average text length of 3,893 characters (755 words)
- Minimum of 91 characters, maximum of 14,957 characters
- Average length of 22 words per sentence
- 48,205 distinct words

➨ The language employed seems to be quite complex

Amazon corpus

Amazon corpus (Custom dataset created from amazon)

- 2,000 reviews written by 1,999 different authors
- Average text length of 341 characters (66 words)
- Minimum of 48 characters, maximum of 3,001 characters
- Average length of 13 words per sentence
- 9,380 distinct words

➨ The language employed seems less complex than the one employed in the IMDb corpus

Twitter corpus

Twitter corpus (Public dataset from sanders analytics)

- Only few tweets were labeled positive or negative
- 1,000 tweets written by an unknown number of authors
- Average text length of 97 characters (15 words)
- Minimum of 9 characters, maximum 140 characters
- Average length of 8 words per sentence
- 3,716 distinct words

➨ The language employed seems rather simple and informal

C-measure

Definition

- Let $g(T, s)$ denote the number of repetitions of a string s in a set of documents T. The C-measure is defined as:

$$C^{\{+,-\}} := \sum_{i=n}^{m} a_{i,n}^{\{+,-\}} \text{ with:}$$

$$a_{i,n}^{\{+,-\}} := \begin{cases} 1, & \text{if } g(T^{\{+,-\}}, c_{i,n}) > 0 \\ 0, & \text{otherwise} \end{cases}$$

C_k-measure

Definition

- Let $g(T, s)$ denote the number of repetitions of a string s in a set of documents T. The C_k-measure is defined as:

$C_k^{\{+,-\}} := \sum_{i=n}^{m} a_{i,n,k}^{\{+,-\}}$ with:

$$a_{i,n,k}^{+} := \begin{cases} 1, & \text{if } g(T^+, c_{i,n}) > k \cdot g(T^-, c_{i,n}) \\ 0, & \text{otherwise} \end{cases}$$

$$a_{i,n,k}^{-} := \begin{cases} 1, & \text{if } g(T^-, c_{i,n}) > k \cdot g(T^+, c_{i,n}) \\ 0, & \text{otherwise} \end{cases}$$

F_k-measure

Definition

- Let $g(T, s)$ denote the number of repetitions of a string s in a set of documents T. The F_k-measure is defined as:

$F_k^{\{+,-\}} := \sum_{i=n}^{m} b_{i,n,k}^{\{+,-\}}$ with:

$$b_{i,n,k}^{+} := \begin{cases} g(T^+, c_{i,n}), & \text{if } g(T^+, c_{i,n}) > k \cdot g(T^-, c_{i,n}) \\ 0, & \text{otherwise} \end{cases}$$

$$b_{i,n,k}^{-} := \begin{cases} g(T^-, c_{i,n}), & \text{if } g(T^-, c_{i,n}) > k \cdot g(T^+, c_{i,n}) \\ 0, & \text{otherwise} \end{cases}$$