University of Insubria
Varese, Italy

Sparse unsupervised feature learning for sentiment classification of short documents

Simone Albertini,
Alessandro Zamberletti, Ignazio Gallo

simone.albertini@uninsubria.it

http://artelab.dista.uninsubria.it

September 23th, 2013

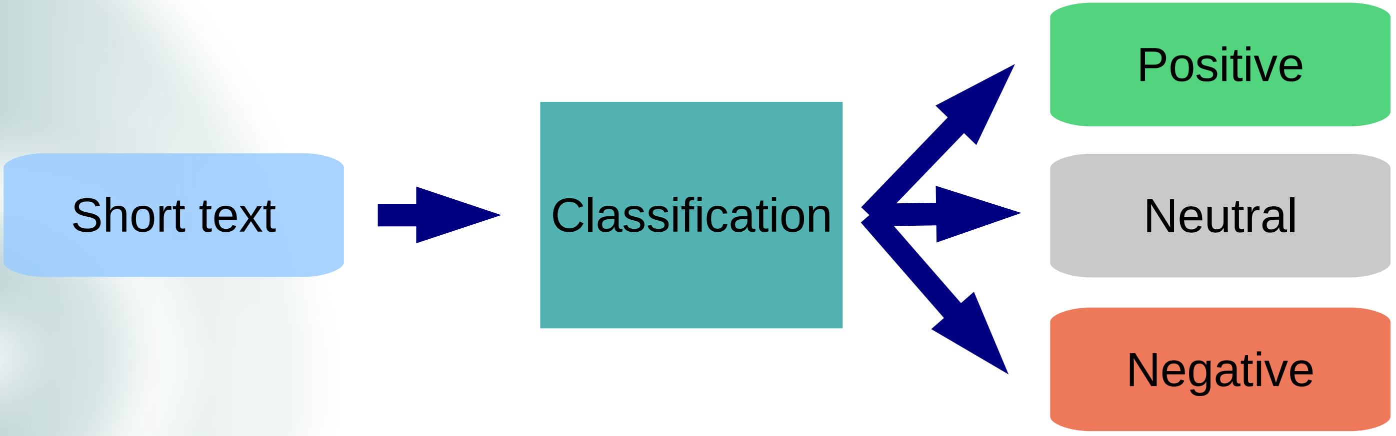
Introducing the problem

- Classification of short texts

  Independent comments

  Phrases from long texts

- Predicting the sentiment polarity

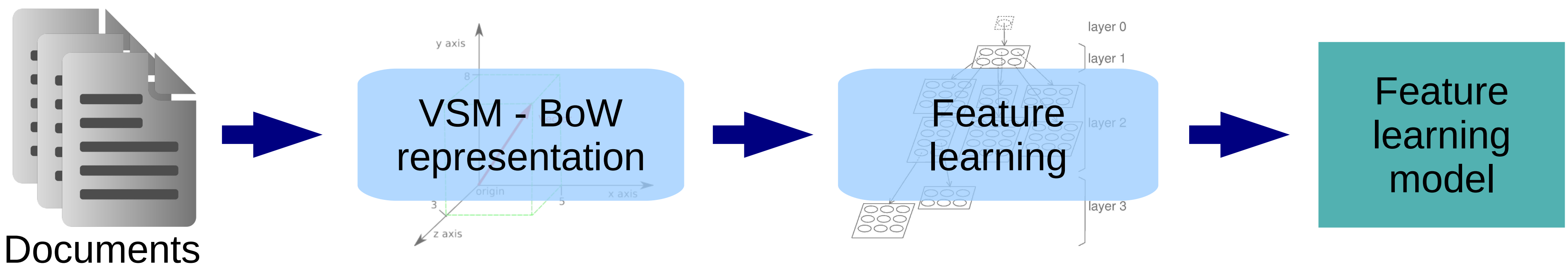


Introducing the problem

- We addressed the problem trying to learn a significative representation of the documents
- *No prior information is used*:
  - No assumpions about language patterns and idioms
  - No opinion-bearing words dictionaries
- The goal is to learn good features starting from several different representations of the documents in a VSM.

Overview of the solution

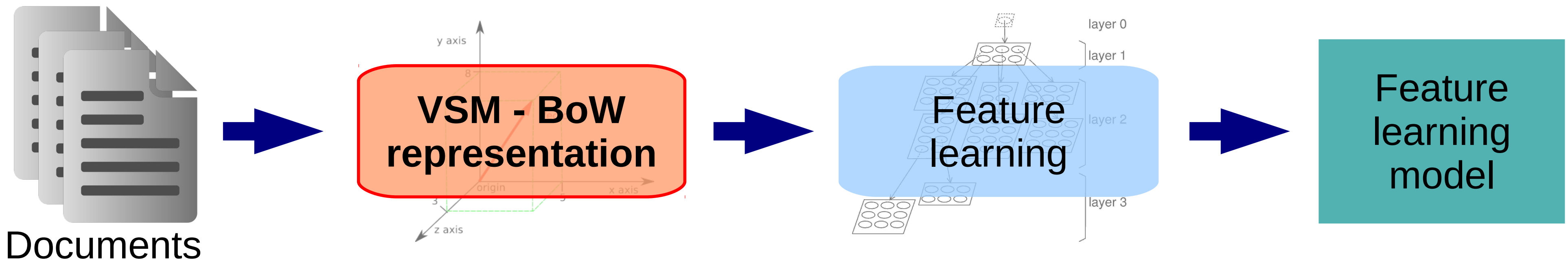
Learning a vector representation



- Unsupervised procedure
- Training a model used to obtain a sparse vector representation of the documents

Overview of the solution

Learning a vector representation



- The documents are represented as vectors
  - Standard Bag of Word approach
  - A dictionary is extracted from the training corpus
  - We tried five differents approaches to compute the scores

Weighting functions

- Binary Term Frequency

$$binary\_score(d,t) = \begin{cases} 1 & \text{if } t \in d \\ 0 & \text{otherwise} \end{cases}$$

- TF-IDF

$$TF \cdot IDF(d,t) = tf(d,t) \cdot \log(\frac{|D|}{df(D,t)})$$

Where:
· *d* is a document
· *t* is a term from the dictionary
· D is the set of all document

Weighting functions

- *Specific against Generic* and *One against All*

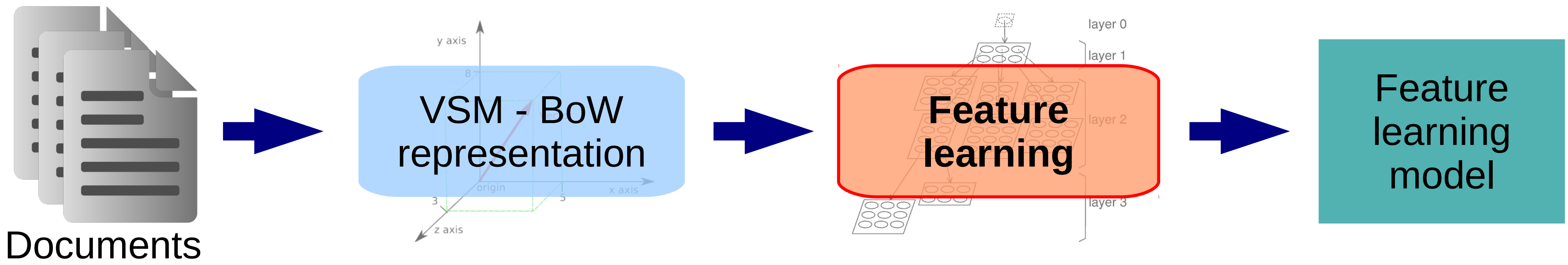
$$score(t, sc, gc) = 1 - \frac{1}{log_2(2 + \frac{F_{t,sc} \cdot D_{t,sc}}{F_{t,gc}})}$$

Where

- *t* Term from the dictionary
- *sc* Specific corpus
- *gc* Generic corpus
- *F* Frequency of a term in a corpus
- *D* Number of document that contains a term in a corpus

	sc	*gc*
Specific against Generic	Positive docs	Negative docs
One against All	All docs	Unrelated docs

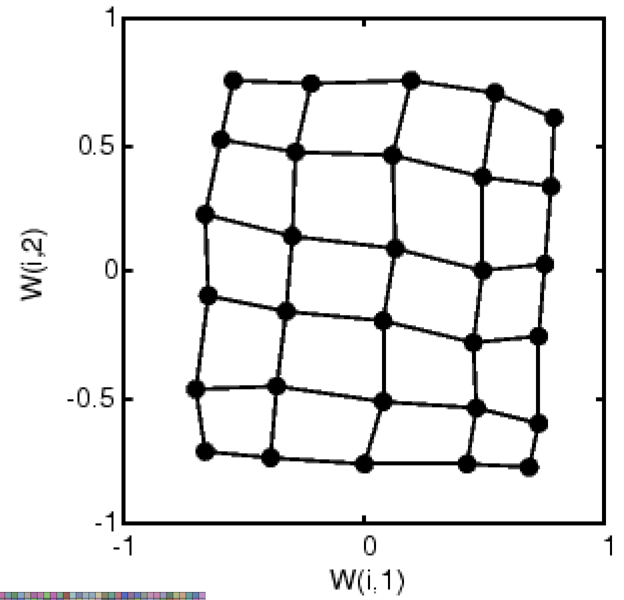
Overview of the solution
Learning a vector representation

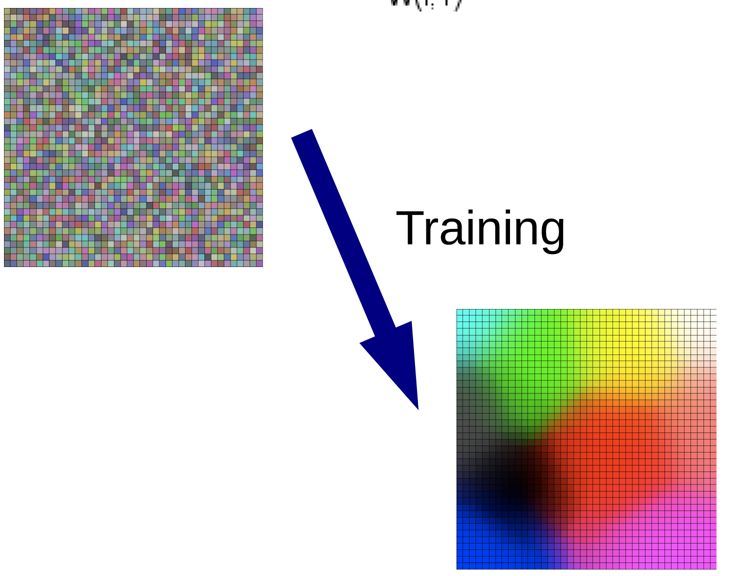


- A *Growing Hyerarchical Self Organizing Map* is used to perform feature learning
  - A GHSOM is an extension of regular 2-dimensional SOMs.

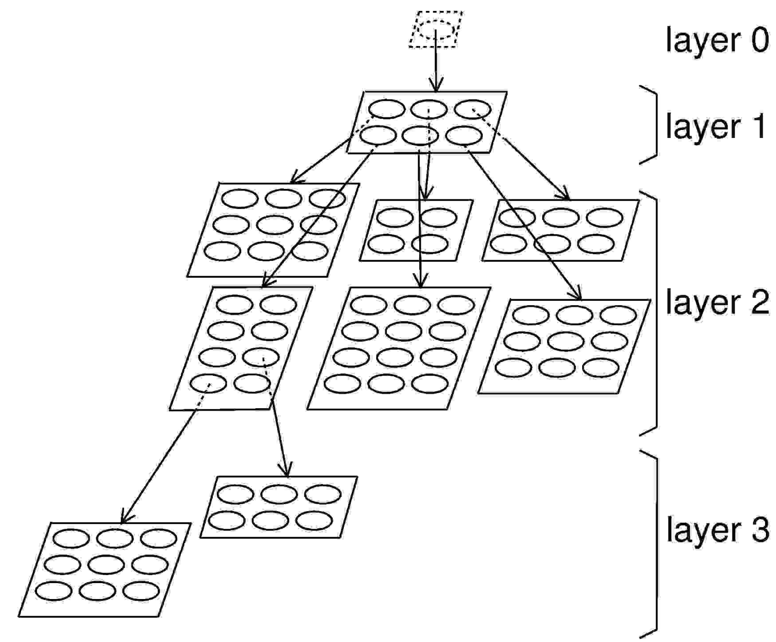
Growing Hyerarchical SOM

- The purpose of a SOM is to learn a quantized representation of the training patterns in their space by adjusting the weights associated to each neuron in order to fit the distribution of the input data.
- It can be considered as a sort of topologically ordered clusterization, where each neuron may represent a cluster whose centroid is given by the vector of the incoming weights.





Growing Hyerarchical SOM

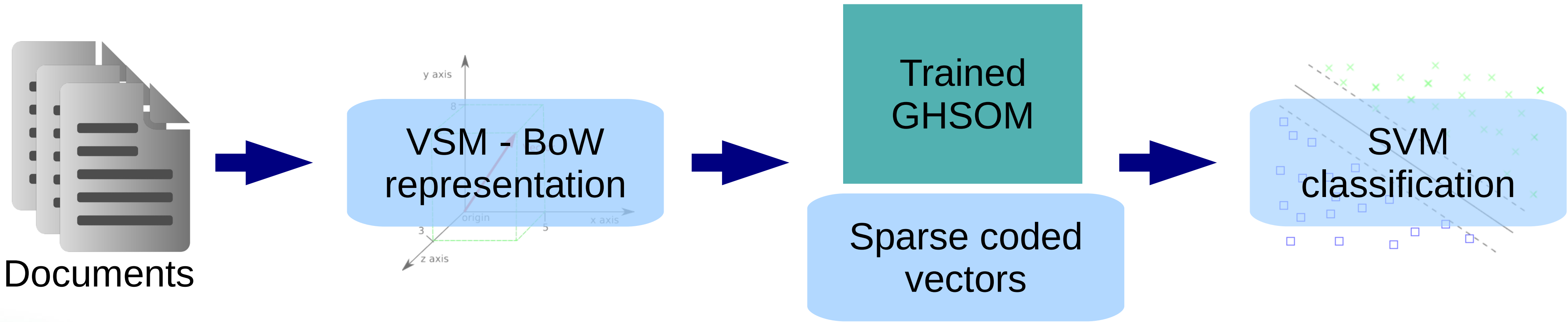


- Two parameters (τ_1, τ_2) control the propensity of the GHSOM to expand in width (for each SOM) and depth respectively.

- The idea is that when the *mean quantization error* of a unit is high, the training algorithm tries to lower it by
  - adding a rows or columns to a SOM (width expansion)
  - Exploding the neuron into another SOM (depth expansion)

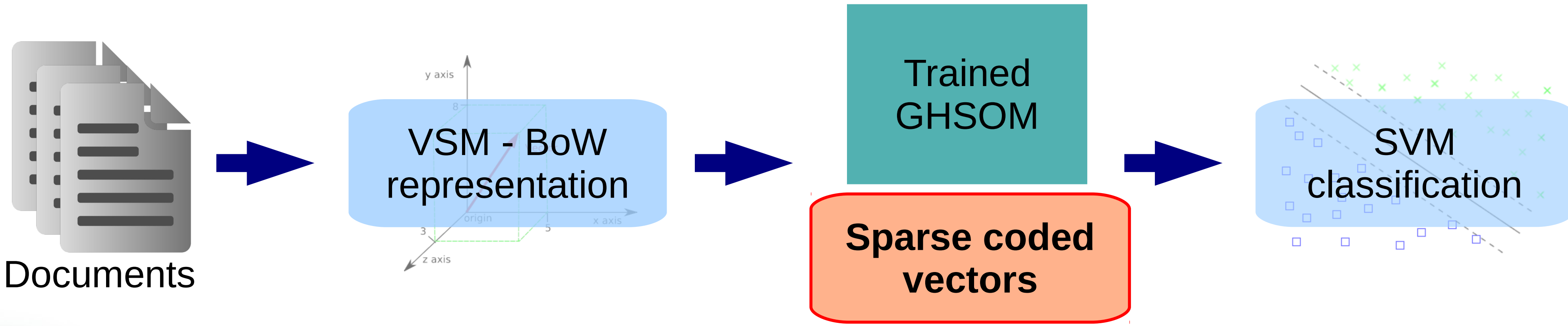
Overview of the solution

Classification of the documents



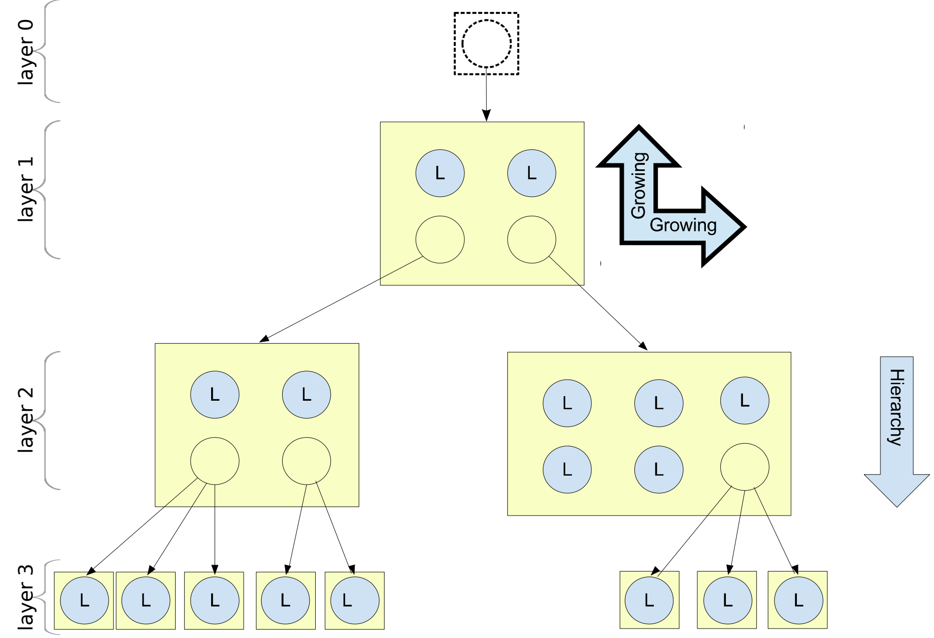
Overview of the solution

Classification of the documents



- The GHSOM is used to map each input vector to a sparse vector in a different space.
    - The starting space has |D| dimension, where D is the size of the dictionary
    - The new space has K dimension, where K is the number of leaves in the GHSOM.

Sparse Vector Representation



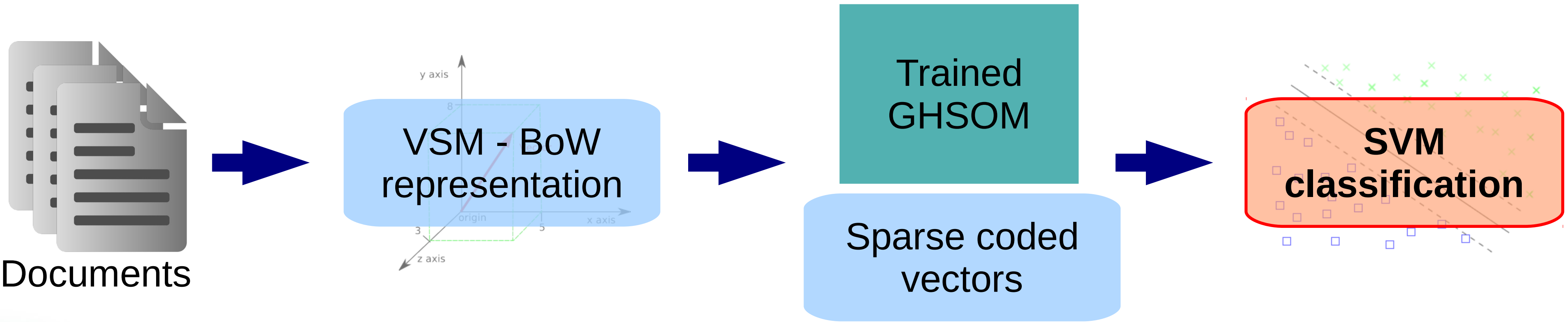
- A leaf unit is a neuron which is not exploded into a new SOM.
- Each leaf unit is assigned a progressive index in [1, K].
- Let *x* be an input vector; x is mapped to a sparse vector f where

$$f(i) = \begin{cases} 1 & \text{if } x \text{ activates } u_i \\ 0 & \text{otherwise} \end{cases}$$

and u_i is the *i*-th leaf unit.

Overview of the solution

Classification of the documents



- Finally, a regular C-SVM is trained to classify the sparse vectors in one of the two classes
  - positive
  - negative

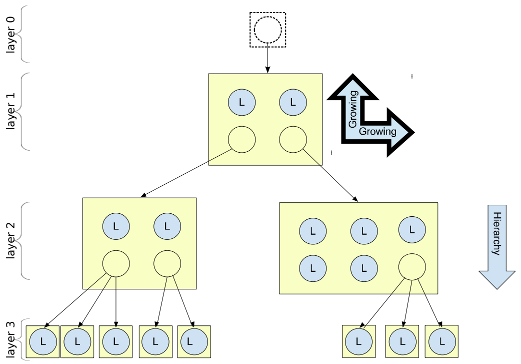
Experiments

- Goals:
  - The contribution of the GHSOM
  - Measure the performances.

- Dataset:
  - Customer review dataset (Hu and Liu, 2004)
    - 1500+ short texts which do not exceed 30 words
    - Annotated short comments about 5 different products
    - It has been balanced

GHSOM analysis

- We assign each leaf unit a polarity label based on majority voting on the polarity of the subset of training patterns quantized by that neuron

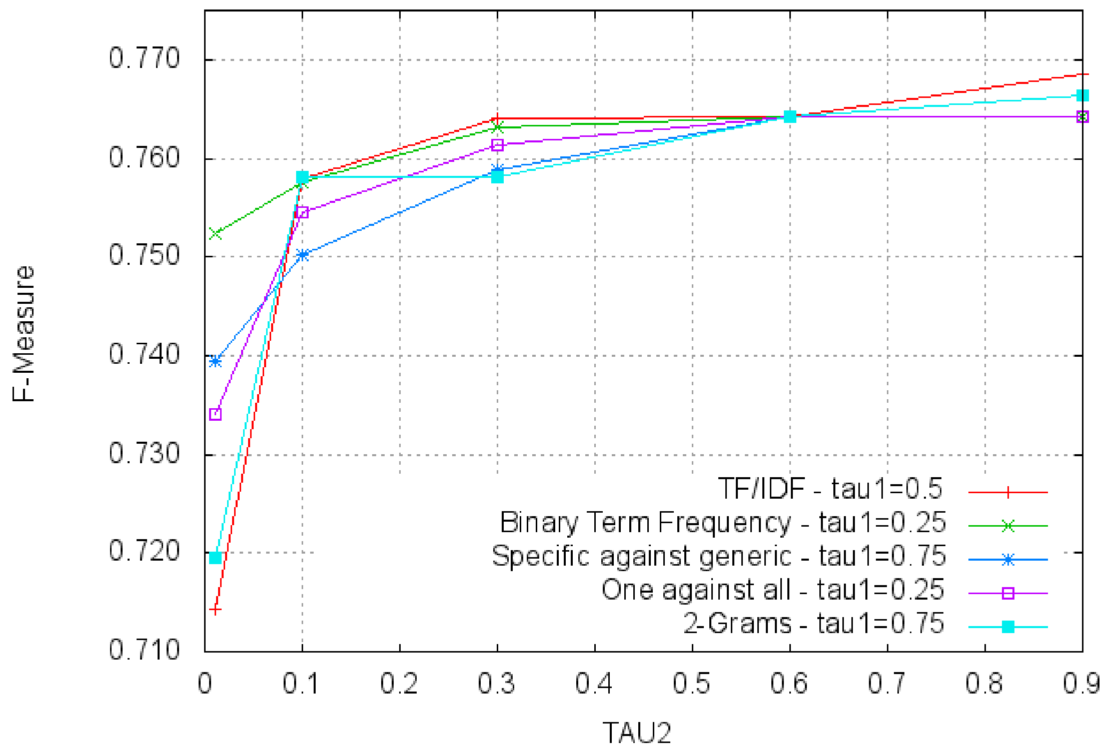


$$pol(u_i) = \begin{cases} pos & \text{if } |P_{pos}| > |P_{neg}| \\ neg & \text{if } |P_{neg}| > |P_{pos}| \\ pol(u_{par}) & \text{otherwise} \end{cases}$$

- Evaluation: classification of the test set by assigning each document the label of the closest neuron.
  - *Here the GHSOM acts like a clusterization algorithm where the neuron's weights are centroids.*

GHSOM analysis

- GHSOM's optimal parameters are found by 5-fold crossvalidation.



NB:

τ_1 → propensity to grow in widht (bigger SOMs)

τ_2 → propensity to grow in depth (more SOMs and more layers)

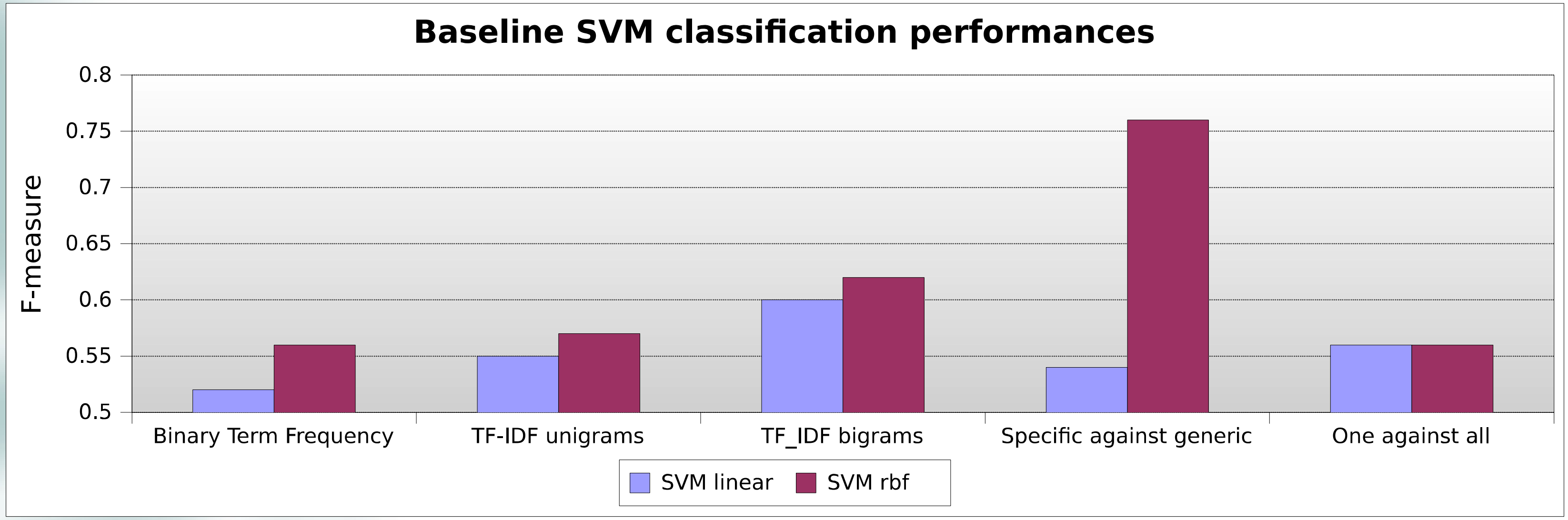
Results

	SVM (baseline)		GHSOM	Full model	
Encoding	**linear**	**RBF**	**analysis**	**linear**	**RBF**
Binary term frequency	0,52	0,56	0,75	0,81	0,87
TF-IDF unigrams	0,55	0,57	0,76	0,76	0,86
TF-IDF bigrams	0,60	0,62	0,76	0,78	0,85
SaG	0,54	0,76	0,76	0,76	0,88
OaA	0,56	0,56	0,77	0,81	0,90

- The table shows the classification results (F-measure)
  - **Baseline**: classification of BoW vectors with no feature learning
  - **GHSOM analysis** (previous slide)
  - **Full model**: classificaiton of the sparse vectors

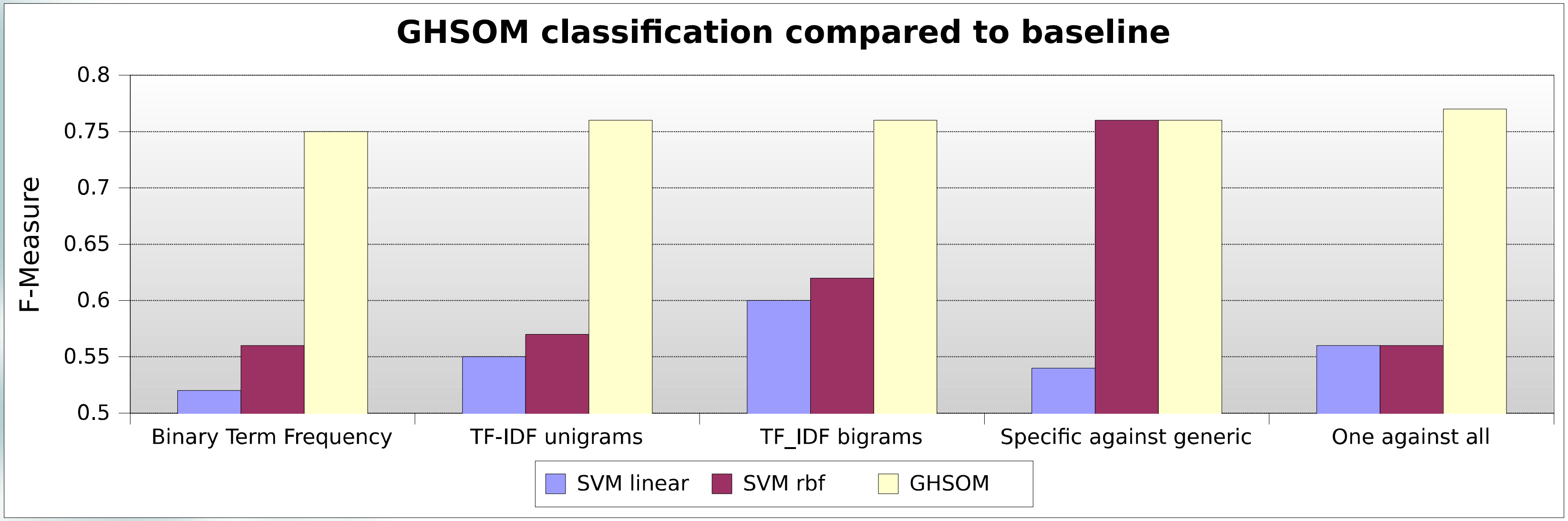
Results

Encoding	SVM (baseline) linear	SVM (baseline) RBF	GHSOM analysis	Full model linear	Full model RBF
Binary term frequency	0,52	0,56	0,75	0,81	0,87
TF-IDF unigrams	0,55	0,57	0,76	0,76	0,86
TF-IDF bigrams	0,60	0,62	0,76	0,78	0,85
SaG	0,54	0,76	0,76	0,76	0,88
OaA	0,56	0,56	0,77	0,81	0,90



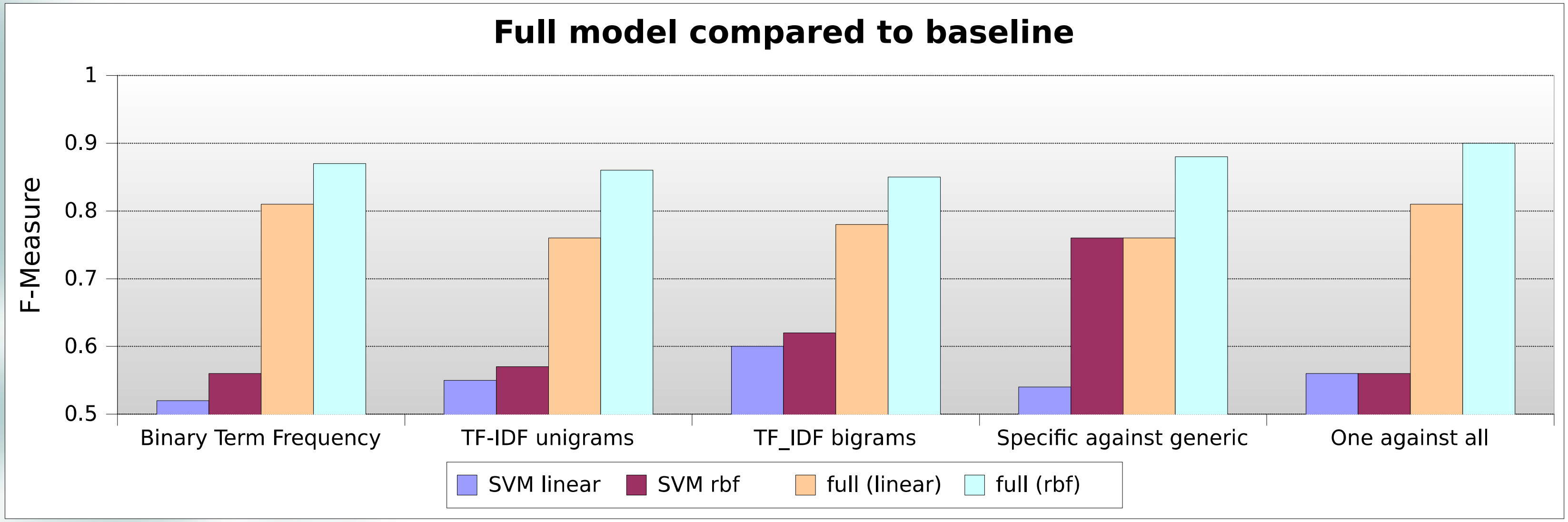
Results

Encoding	SVM (baseline) linear	SVM (baseline) RBF	GHSOM analysis	Full model linear	Full model RBF
Binary term frequency	0,52	0,56	0,75	0,81	0,87
TF-IDF unigrams	0,55	0,57	0,76	0,76	0,86
TF-IDF bigrams	0,60	0,62	0,76	0,78	0,85
SaG	0,54	0,76	0,76	0,76	0,88
OaA	0,56	0,56	0,77	0,81	0,90



Results

Encoding	SVM (baseline) linear	SVM (baseline) RBF	GHSOM analysis	Full model linear	Full model RBF
Binary term frequency	0,52	0,56	0,75	0,81	0,87
TF-IDF unigrams	0,55	0,57	0,76	0,76	0,86
TF-IDF bigrams	0,60	0,62	0,76	0,78	0,85
SaG	0,54	0,76	0,76	0,76	0,88
OaA	0,56	0,56	0,77	0,81	0,90



Conclusions

- This is an experiment using a novel feature learning method.
- It proves that a *feature learning* approach outperforms standard BoW representations.
- Generally, it is my opinion that the correct way to solve a classification task is to automatically learn features rather then fixing them.
- Shift the effort from “*hand craft good features*” to “*correctly learn good features*”.