Mapping of author created learning objectives to official common academic goals

Gandalf Saxe, Jacob J. Hansen, Jakob D. Havtorn, Yevgen Zainchkovskyy

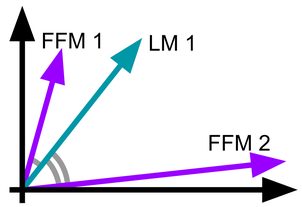
Introduction

The Danish publisher Gyldendal, seeks to classify learning objectives defined by their content authors (LMs) according to the Common Academic Goals (FFMs, Fælles Faglige Mål) defined by the Ministry of Education. This classification must be unsupervised since only very few labels are available. The goal is to develop a model that can help Gyldendal obtain a labelled data set by live recommending a set of FFMs to the authors as they write their LM.

Model 1: Latent Semantic Indexing (LSI a.k.a LSA) [3]

Basic bag of words model. *X* is a matrix with sentences mapped to columns, words mapped to rows and the number of occurrence of a word in a sentence as the elements. The matrix product XX^T contains the correlations between words t_i and t_j in the form of the dot products $t_i^T t_j$. From this, obtain the rank-*k* approximation to XX^T by selecting the *k* largest singular values. Corresponds to PCA applied to *X*.

$$\mathbf{t}_i^T \rightarrow \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix} = X = U\Sigma V^T \quad \text{with } \mathbf{d}_j \downarrow$$



Then compare sentences *i* and *j* by

$$\text{cosine similarity}(\Sigma_k \mathbf{d}_i, \Sigma_k \mathbf{d}_j)$$

Model 2: FastText [1,2,4]

Facebook's implementation of word2vec by skip-gram negative sampling.

The FastText summary to the right describes the basic skip-gram idea, as well as the three **enhancement-tricks** and three **speed-up tricks**.

Training:

The model is a shallow neural network with one hidden layer. The words are converted into a bag of n-grams (see [TRICK2]) and fed one-hot encoded into the input layer. We give a target word, and want to predict the context (neighbouring words). **This makes it possible to encode word semantics *and* syntactics in a vector, unsupervised on a large corpus of text.** The hidden layer weight matrix becomes a lookup table of vectors for different words (or rather bag of n-grams).

Classification:

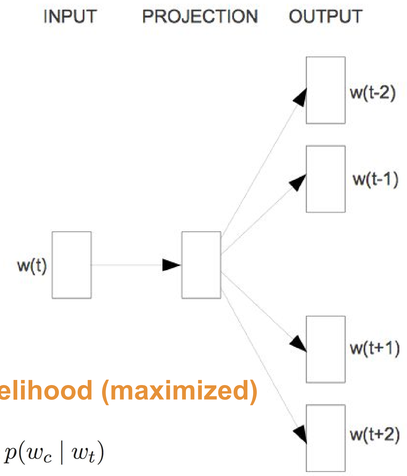
1. Input: word. **2. Conversion** to bag of n-grams. **3. Intermediate result:** Vector representation of word, here a 300 dimensional vector. **4. Output:** Probability distribution of all words in vocab to be context word. **5. How we instead get output:** we simply take cosine similarity between two sentences. A sentence is basically represented by the sum of the individual words. Hence, the correlation between an FFM and any given LM is computed by the angular distance

Model 2.A: FastText with naïve online learning

All known mappings are included in FFM database and point to their respective FFMs, thus artificially boosting the FFM database size, performance est. with 5-fold CV

FastText summary

1. Word2vec skip-gram neural network:



2. Log-likelihood (maximized)

$$\sum_{t=1}^{T} \sum_{c \in \mathcal{C}_t} \log p(w_c \mid w_t)$$

3. Basic p(wc|wt) given by softmax (for some scoring function s):

$$p(w_c \mid w_t) = \frac{e^{s(w_t,\, w_c)}}{\sum_{j=1}^{W} e^{s(w_t,\, j)}}$$

4. We approximate p(wc|wt) by negative sampling (only update small sample [TRICK1]/[SPEED1]:

$$\log(1 + e^{-s(w_t,\, w_c)}) + \sum_{n \in \mathcal{N}_{t,c}} \log\left(1 + e^{s(w_t,\, n)}\right)$$

5. w(t) is broken into many n-grams [TRICK2]:

Subword example ("tidsalder")

Word: tidsalder

2-grams: ti, id, ds, al, ld, de, er

3-grams: **tid**, ids, dsa, sal, ald, lde, der

4-grams: **tids**, idsa, dsal, sald, alde, lder

5-grams: tidsa, idsal, dsald, salde, **alder**,

6-grams: tidsal, idsald, dsalde, salder

→ Tidsalder: {index, (hash of all n-grams)}

..and have scoring function:

$$s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^\top \mathbf{v}_c$$

$\mathcal{G}_w$ is the set of n-grams in word w.

Each n-gram g is associated with vector $\mathbf{z}_g$

6. Subsampling of frequent words [TRICK3/SPEED2]

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

$P(w_i)$ = discard prob. of word w_i for some threshold t

7. n-grams stored in hashtable [SPEED3]

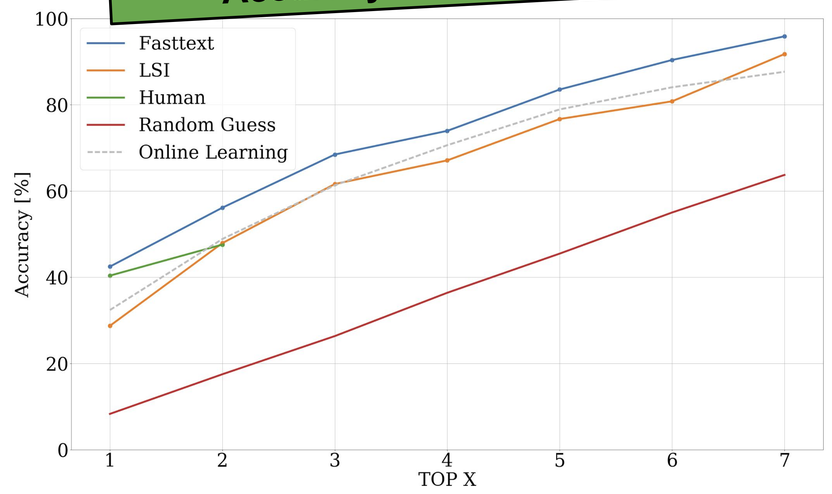
8. No n-grams for P most frequent words [SPEED4]

Note: P = words in vocabulary → Ordinary skip-gram

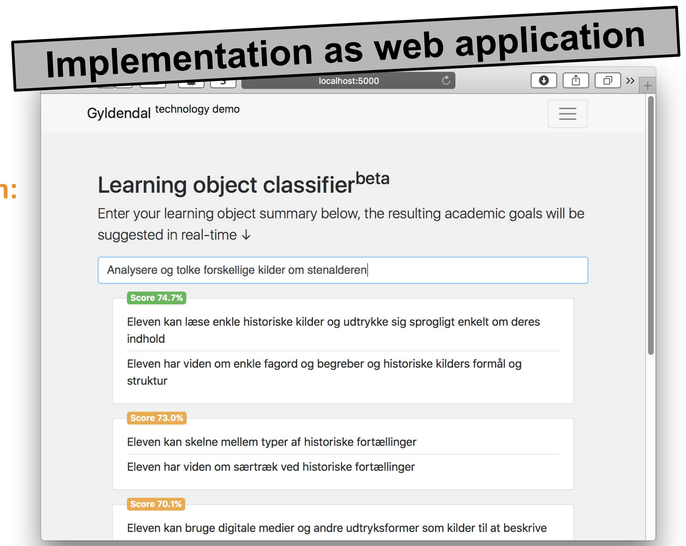
Author vs Fælles Faglige Mål

LM 1	→	FFM 1
LM 2	→	FFM 2
...	→ ?	...
LM N	→ ?	FFM M

Accuracy of LSI and FastText

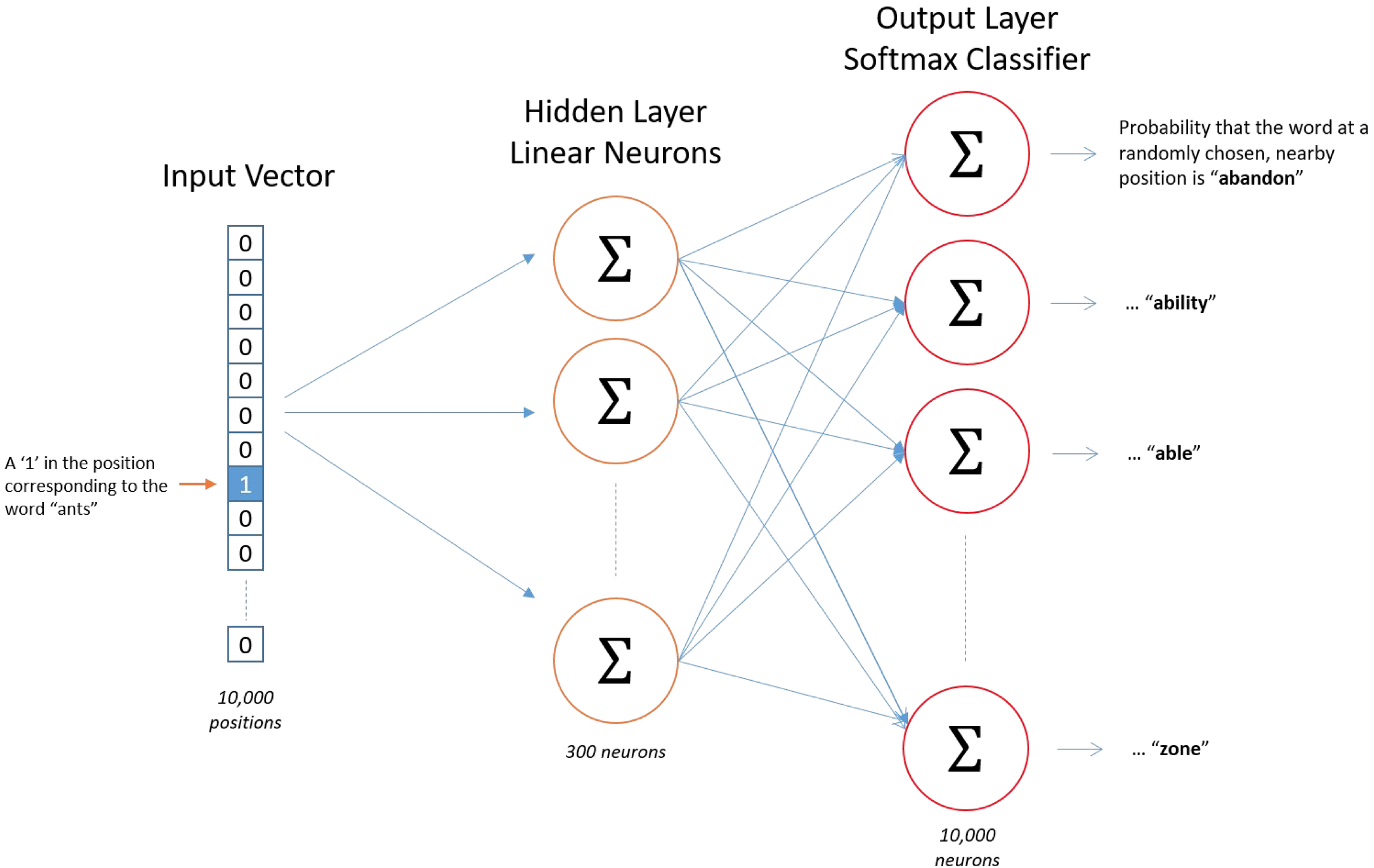


Implementation as web application

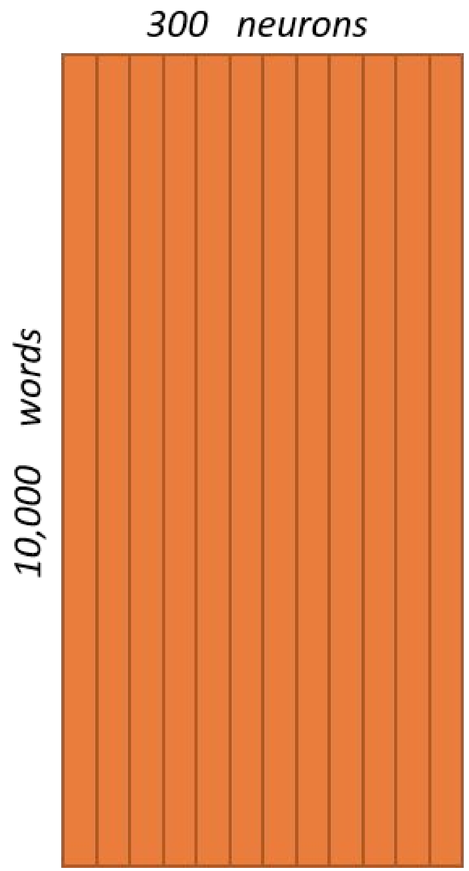


[1] Murphy, Kevin P. (2012). Machine Learning A Probabilistic Perspective
[2] Mikov, Tomas et al (2013). Distributed Representations of Words and Phrases and their Compositionality
[3] Mikov, Tomas et al (2016). Efficient Estimation of Word Representations in Vector Space
[4] Joulin, Armand et al (2016). Bag of Tricks for Efficient Text Classification

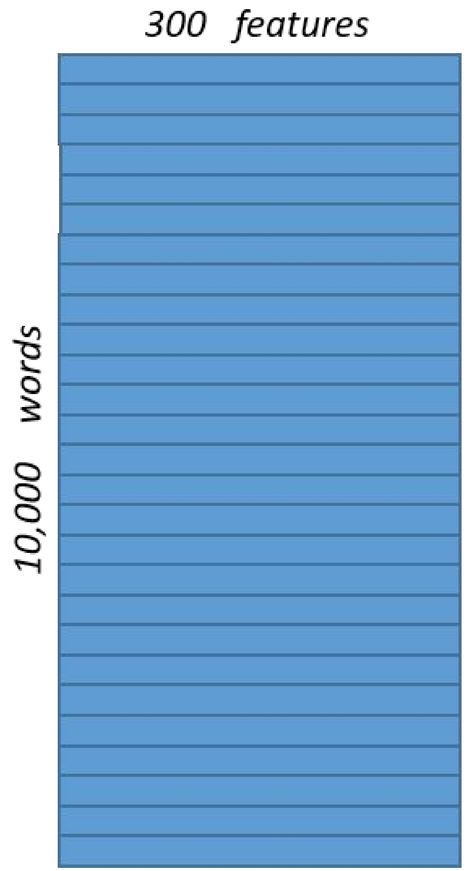
Source Text	Training Samples
The quick brown fox jumps over the lazy dog.	(the, quick) (the, brown)
The **quick** brown fox jumps over the lazy dog.	(quick, the) (quick, brown) (quick, fox)
The quick **brown** fox jumps over the lazy dog.	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown **fox** jumps over the lazy dog.	(fox, quick) (fox, brown) (fox, jumps) (fox, over)


Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
0
0
0
0
0
0
1
0
0
0
A '1' in the position corresponding to the word "ants"
10,000 positions
300 neurons
10,000 neurons
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"

Hidden Layer
Weight Matrix



Word Vector
Lookup Table!



Output weights for “car”

Word vector for “ants”

×

softmax

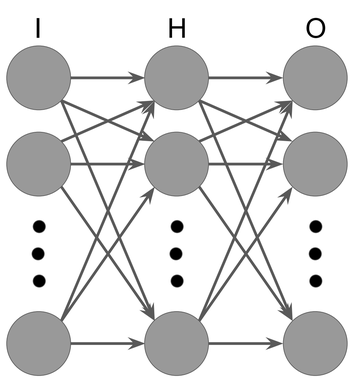
$$\frac{e^x}{\sum e^x}$$

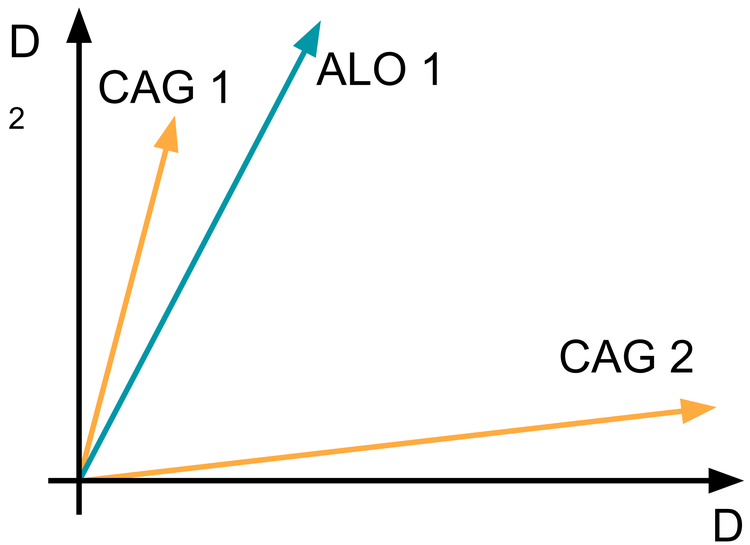
=

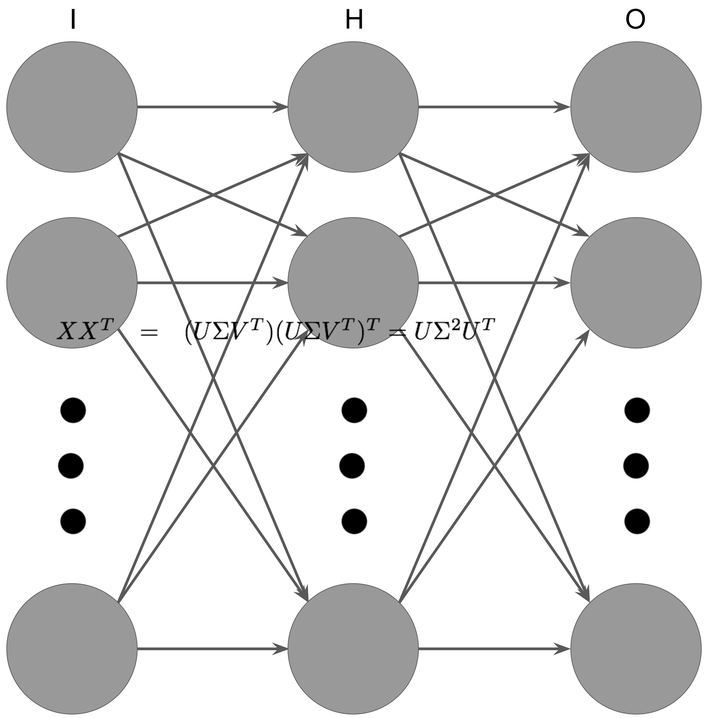
Probability that if you randomly pick a word nearby “ants”, that it is “car”

Author vs Fælles Faglige Mål

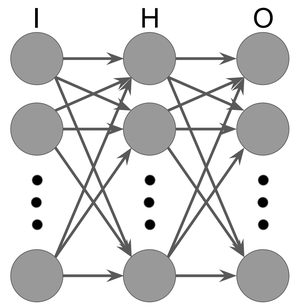
LM 1		FFM 1
LM 2		FFM 2
...	?	...
LM N	?	FFM M




D
2
CAG 1
ALO 1
CAG 2
D
1



$$XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma^2 U^T$$



Mapping of author created learning objectives to official common academic goals

Gandalf Saxe, Jacob J. Hansen, Jakob D. Havtorn, Yevgen Zainchkovskyy

Introduction
The Danish publisher Gyldendal, seeks to classify learning objectives defined by their content authors (ALOs) according to the Common Academic Goals (CAGs, Fælles Faglige Mål) defined by the Ministry of Education. This classification must be unsupervised since only very few labels are available. The goal is to develop a model that can aid Gyldendal in obtaining a labelled data set by making live recommendations of a set of CAGs to the authors as they are writing their ALO,

Latent semantic analysis (LSA) [3]
X is a matrix with sentences in its columns, words in its rows and the number of occurrence of a word in a sentence as the elements. The matrix product XX^T contains the correlations between words t_i and t_j in the form of the dot products $t_i^T t_j$. From this, obtain the rank-*k* approximation to XX^T by selecting the *k* largest singular values. Corresponds to PCA applied to *X*.

$$\mathbf{t}_i^T \rightarrow \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix} = X = U\Sigma V^T \qquad XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma^2 U^T$$

$\mathbf{d}_j \downarrow$

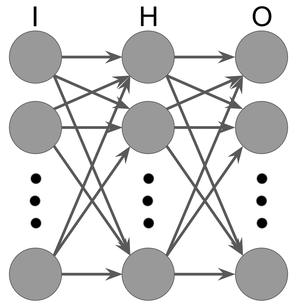
Then compare sentences *i* and *j* by

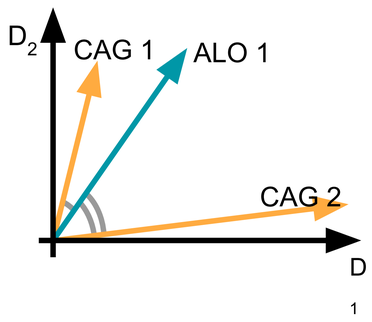
$$\text{cosine similarity}(\Sigma_k \mathbf{d}_i, \Sigma_k \mathbf{d}_j)$$

FastText [1,2]
Training:
Uses a shallow neural network with one hidden layer. The input layer has words represented by 1 of V. Skip-gram model uses neighbouring words (context) to predict center word. This makes it possible to train supervised on a large corpus of text. Very fast training achieved through shallow architecture and efficient datastructure for storing vocabulary (Huffman Binary Tree).

Classifying:
Given a word, the FastText model gives the learned vector representation. Here, this representation is a 300 dimensional vector. A representation for a sentence is obtained by summing the representations of the individual words. The correlation between a CAG and any given ALO is computed by the angular distance

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2}, \quad \bar{\theta}_{[0,1]} = \frac{\theta}{\pi}$$



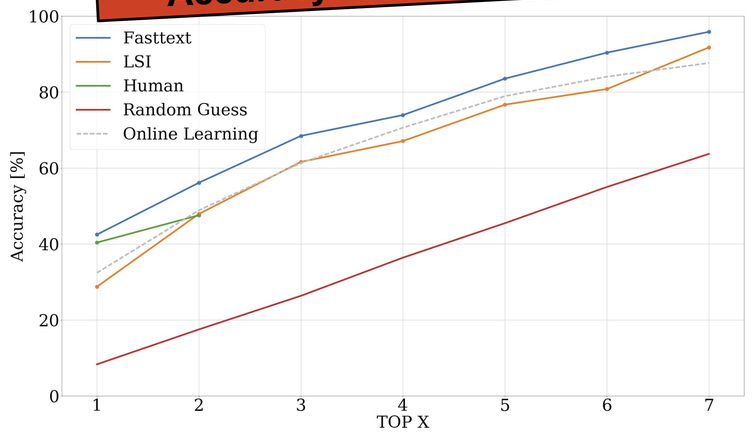


FastText with online learning
By considering similarity to already known mappings as new pointers to CAGs, an additional, online, semi-supervised model was evaluated. Accuracy of the 2-fold hold-out cross validated model is shown to the right.

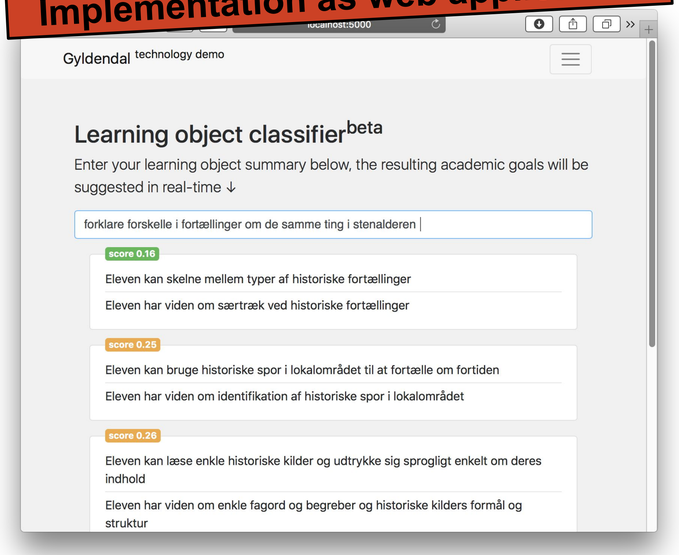
Author vs Common Learning Objective

ALO 1	→	CAG 1
ALO 2	→	CAG 2
...	→ ?	...
ALO N	→ ?	CAG M

Accuracy of LSA and FastText



Implementation as web application



[1] Joulin, Armand et al (2016). Bag of Tricks for Efficient Text Classification
[2] Mikolov, Tomas et al (2016). Distributed representations of words and phrases and their compositionality
[3] Murphy, Kevin P. (2012). Machine Learning A Probabilistic Perspective