Seminar 4

Definition 1 (Inverse document frequency)
Inverse document frequency of a term t is defined as

$$idf_t = \log\left(\frac{N}{df_t}\right)$$

where N is the number of all documents and df_t (the document frequency of t) is the number of documents that contain t.

Definition 2 (tf-idf weighting scheme)
In the tf-idf weighting scheme, a term t in a document d has weight

$$tf\text{-}idf_{t,d} = tf_{t,d} \cdot idf_t$$

where $tf_{t,d}$ is the number of tokens t (the term frequency of t) in a document d.

Definition 3 (ℓ^2 (cosine) normalization)
A vector v is cosine-normalized by

$$v_j \leftarrow \frac{v_j}{||v||} = \frac{v_j}{\sqrt{\sum_{k=1}^{|v|} {v_k}^2}}$$

where v_j is the element at the j-th position in v.

Definition 4 (Sublinear term frequency scaling)
The weight of a term t in a document d is determined as

$$w_{t,d} = \begin{cases} 1 + \log\left(tf_{t,d}\right) & \textit{if } tf_{t,d} > 0 \\ 0 & \textit{otherwise} \end{cases}$$

where $tf_{t,d}$ is the number of tokens t (the term frequency of t) in a document d.

Exercise 4/1

Consider the frequency table of the words of three documents doc_1, doc_2, doc_3 below. Calculate the *tf-idf* weight of the terms *car, auto, insurance, best* for each document. *idf* values of terms are in the table.

	doc_1	doc_2	doc_3	idf
car	27	4	24	1.65
auto	3	33	0	2.08
insurance	0	33	29	1.62
best	14	0	17	1.5

Table 1: Exercise.

Exercise 4/2

Count document representations as normalized Euclidean weight vectors for each document from the previous exercise. Each vector has four components, one for each term.

Exercise 4/3

Based on the weights from the last exercise, compute the relevance scores of the three documents for the query *car insurance*. Use each of the two weighting schemes:

a) Term weight is 1 if the query contains the word and 0 otherwise.

b) Euclidean normalized *tf-idf*.

Please note that a document and a representation of this document are different things. Document is always fixed but the representations may vary under different settings and conditions. In this exercise we fix document representations from the last exercises and will count relevance scores for query and documents under two different representations of the query. It might be helpful to view on a query as on another document, as it is a sequence of words.

Exercise 4/4

Consider a collection of documents and the terms *dog*, *cat* and *food* that occur in 10^{-3x}, 10^{-2x} and 10^{-x} of the documents, respectively. Now document doc1 contains the words $2y$, y and $3y$ times and doc2 $2z$, $3z$ and z times. Order these two documents based on vector space similarity with the query *dog food*.

Exercise 4/5

Calculate the vector-space similarity between the query *digital cameras* and a document containing *digital cameras and video cameras* by filling in the blank columns in the table below. Assume $N = 10000000$, sublinear term frequency scaling from Definition 4 (columns w) for both query and documents, *idf* weighting only for the query and cosine normalization only for the document. *and* is a STOP word.

		Query				Document			relevance
	df	*tf*	w	*idf*	q	*tf*	w	d	$q \cdot d$
digital	10 000								
video	100 000								
cameras	50 000								

Table 2: Exercise.

Exercise 4/6

Show that for the query $q_1 =$ *affection* the documents in the table below are sorted by relevance in the opposite order as for the query $q_2 =$ *jealous gossip*. Query is *tf* weight normalized.

	SaS	PaP	WH
affection	0.996	0.993	0.847
jealous	0.087	0.120	0.466
gossip	0.017	0	0.254

Table 3: Exercise.