Practical AI: NLP. Semantics part 2

Stanislav Protasov for
Harbour.Space University

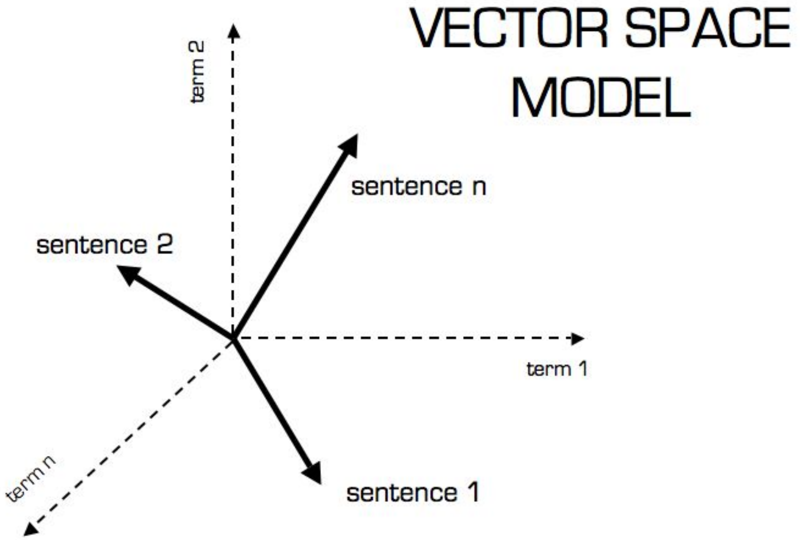
Before we start…

Small challenge!

TF-IDF

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Search engine



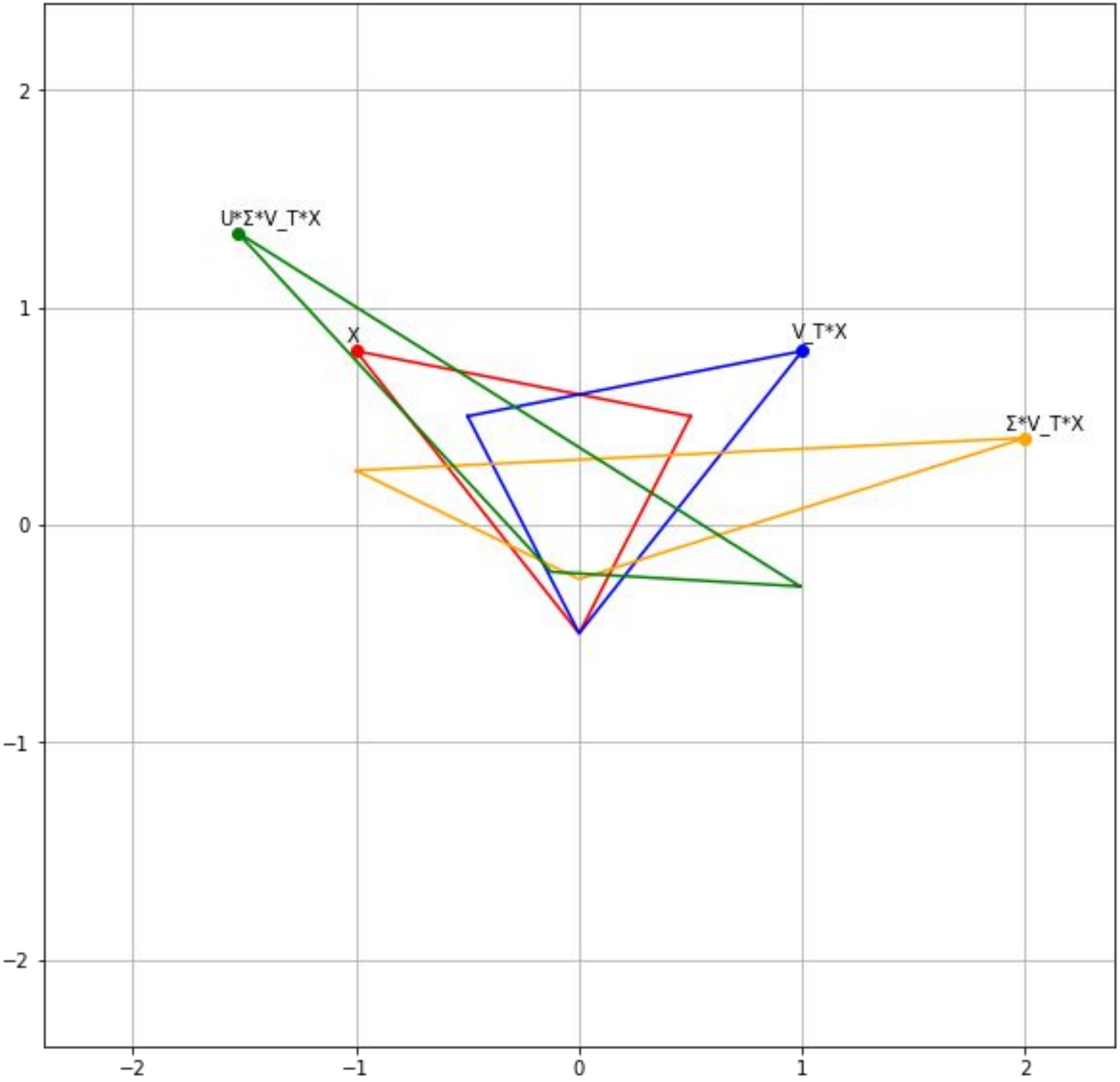
$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Documents to vectors (LSA)

1) Build terms-document matrix
2) Reduce dimensions (LSA) preserving similarity measure
3) Profit!

Latent semantic analysis can be easily performed using **PCA** (principal component analysis) which can be performed using **SVD** (singular value decomposition) of terms-document matrix. Or other algorithm :)

$$\underset{m \times n}{\overset{X}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m \times r}{\overset{U}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}}} \underset{r \times r}{\overset{S}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}}} \underset{r \times n}{\overset{V^{\mathsf{T}}}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}}$$


U*Σ*V_T*X
X
V_T*X
Σ*V_T*X
2
1
0
−1
−2
−2
−1
0
1
2

Lab #1. Document to smaller vector (reading)

Study this example https://github.com/str-anger/hsu.ai/blob/master/code/05.%20SVD%20and%20PCA%20magic.ipynb

Apply provided techniques to reduce number of dimensions in term-document matrix.

What do you need to run search engine?

Considering context: word2vec

CBoW (Continuous bag of words) - predict a word given a context

N-skip-grams - predict a context given a word

In both models word order doesn't matter.

This models are trained in **reduced** space.

Word2vec is a tool to train such models.

Deep Structured Semantic Model (DSSM) - cooler version of semantic analysis from Microsoft.

There are also **sent2vec**, **text2vec**, ...

Homework #1: replace PCA in your search engine with doc2vec

PCA considers **text as a bag of words**. For short texts this works ok, but for longer texts it doesn’t catch the difference between “A killed B” and “B killed A”, although it encodes the fact of murder.

*2vec methods consider word appearances in relatively small surrounding, that brings order into context. Advances methods like DSSM also work with 3-trams.

Your hometask is to sum up results of today’s labs and build **search engine** powered with **doc2vec** technology.

Lab #2. Embedding with doc2vec.

Solve at least 1&2 out of 3.
https://github.com/hsu-ai-course/hsu.ai/blob/master/code/05.%20NLP%20Semantics%20with%20word2vec%20and%20doc2vec.ipynb

1. Train doc2vec using “war and peace” sentences.
2. Write a function that embed a string using a model created.
3. (*) Implement search engine for Jeopardy questions.

Machine translation today

Companies move from distributional models to more accurate **semantic models**.

Semantics is **shared among languages**.

See example to try machine translation.

Lab #3. Machine translation

1. Obtain developer’s key at https://translate.yandex.com/developers
2. Run this code https://github.com/hsu-ai-course/hsu.ai/blob/master/code/05.%20Machine%20translation.ipynb

Hometask: Speak with AI in your language

1) Write a search engine that accepts queries in Spanish, but can search texts in English.
2) Build a database for https://github.com/hsu-ai-course/hsu.ai/blob/master/code/datasets/nlp/facts.txt: stem words, prepare TDM using doc2vec.
3) Implement search algorithm similar to hometask 05.
4) Add translation of queries **es->en**.
5) Test your solution for queries:
    a) ¿Por qué las nutrias de mar se dan la mano así?
    b) ¿Dime algo sobre los gorilas?