Unsupervised learning for news summarization

how to automatically extract the most important information

Jakub Kubajek

Agenda

1. Introduction
2. Words importance
3. Embeddings
4. Clustering
5. Summarization

Introduction

Why unsupervised learning?

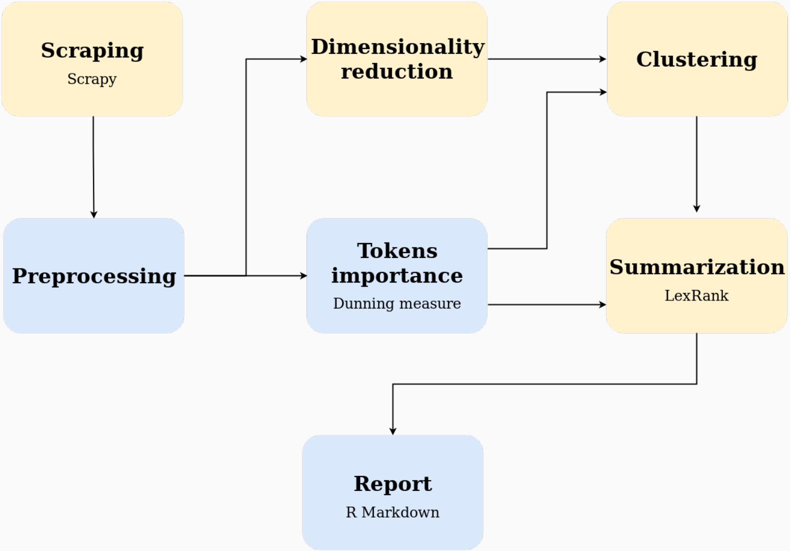
- Lack of pre-trained models for **Polish**
- Shortage of training data in Polish
- Low computing resources
- Multi-document problem - hundreds articles every day

Definition of key words

- **Token** - word
- **Lemmatization** - reducing word to its base grammar form
- **TF matrix** - matrix that describes the frequency of terms occuring in a collection of documents
- **IDF** - Inverse Document Frequency
- **Embedding** - numeric (vector) representation of a word
- **Cosine similarity** - a measure of similarity of two vectors

$$cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|}$$

Model's overview



Words importance

Dunning’s measure

General information

- Measures whether token’s **frequency** in a particular day is statistically different from that in a reference period
- Likelihood ratio test
- No normality assumption - **binomial** distribution
- Proper measure for **rare** events

Dunning's measure

General information

- Measures whether token's **frequency** in a particular day is statistically different from that in a reference period
- Likelihood ratio test
- No normality assumption - **binomial** distribution
- Proper measure for **rare** events

Equation

$$\lambda = \frac{L(p, k_0, n_0)\, L(p, k_1, n_1)}{L(p_0, k_0, n_0)\, L(p_1, k_1, n_1)} \tag{1}$$

$$L(p, k, n) = p^k (1-p)^{n-k} \tag{2}$$

where, $p = \frac{k_0+k_1}{n_0+n_1}$, $p_i = \frac{k_i}{n_i}$ and **-2*log*λ** has χ^2 distribution with one degree of freedom. k_0 is the number of word's occurrences **without** a particular day, k_1 is the word's count **in** a particular day.

Model's implementation

Modification of Dunning measure

- Multiplication by -1, when $p_1 < p_0$
- Counteracting logarithm of zero: $p = max(eps, p)$

Model's implementation

Modification of Dunning measure

- Multiplication by -1, when $p_1 < p_0$
- Counteracting logarithm of zero: $p = max(eps, p)$

Selecting words for clustering

- $-2log\lambda \geq 10$
- Token count (k_i) larger than the value of 90^{th} percentile of all counts

Embeddings

Latent Semantic Analysis (LSA)

- Singular Value Decomposition (**SVD**) of TF matrix composed of both **paragraphs** and **articles**
- No need for training - algebra
- Dimensionality reduction
- Capturing **semantic** relation between words

Clustering

Clustering

Agglomerative hierarchical clustering

- Start with N topics
- Find two the most similar topics - **cosine** between embeddings
- Merge the two topics - set new embedding as a **sum** of the two
- Stop when there is 1 topic
- Return clustering optimal according to the **silhouette** algorithm

Silhouette algorithm

It aims to find clusters such that objects inside groups are the most **similar** to each other and **dissimilar** to objects from other clusters.

Equation

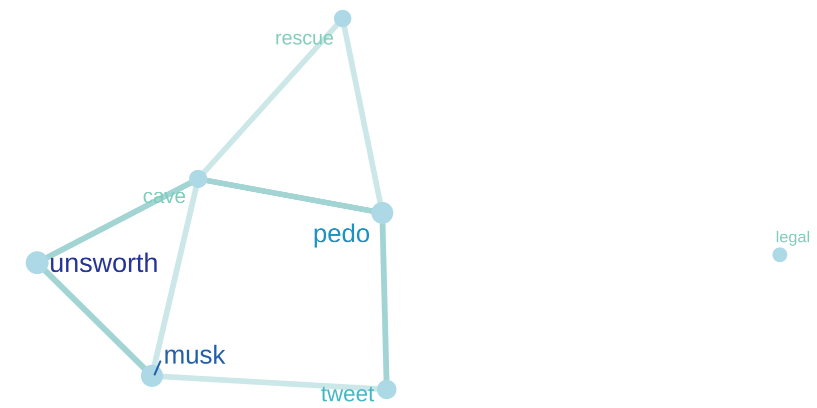
$$a(i) = cos(e_i,\, e_{C_k} - e_i)$$

$$b(i) = \min_{l \neq k} cos(e_i,\, e_{C_l})$$

$$s(i) = \begin{cases} \frac{b(i)-a(i)}{max\{b(i), a(i)\}}, & \text{if } |C_k| > 1 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $a(i)$ and $b(i)$ are inner and outer similarity, e_i is an embedding of i^{th} token, e_{C_k} is the topic's embedding where $i \in C_k$.

Example



Summarization

LexRank

An extractive summarization algorithm, based on Google's **PageRank**, aiming to select sentences that are highly linked (by common words) to other highly linked sentences.

- PageRank
  - Used to rank web pages
  - Ranking equal to the **steady state** of Markov chain
  - Markov matrix obtained as a weighted mean of normalized matrix of **links** between pages and **random** matrix ($\frac{1}{N}$ in every cell)
- LexRank
  - Cosine similarity between sentences
  - Uses **TF-IDF** matrix

Implementation in the model

Modifications

- Articles filtering - minimal topic words frequency

Implementation in the model

Modifications

- Articles filtering - minimal topic words frequency
- Weights of words
  - **TF**
  - Modified **Dunning** measure
    - $log(D_i + 1)$ when $p_1 \geq p_0$
    - 0 when $p_1 < p_0$
  - Cosine **similarity** between word and topic embeddings

Implementation in the model

Modifications

- Articles filtering - minimal topic words frequency
- Weights of words
  - **TF**
  - Modified **Dunning** measure
    - $log(D_i + 1)$ when $p_1 \geq p_0$
    - 0 when $p_1 < p_0$
  - Cosine **similarity** between word and topic embeddings
- Sentences' **embeddings**

Implementation in the model

Modifications

- Articles filtering - minimal topic words frequency
- Weights of words
  - **TF**
  - Modified **Dunning** measure
    - $log(D_i + 1)$ when $p_1 \geq p_0$
    - 0 when $p_1 < p_0$
  - Cosine **similarity** between word and topic embeddings
- Sentences' **embeddings**
- Ranking scaling - $F_i = \frac{log(w_i^T)}{log(w_i)}$
  - where F_i is a scaling factor, w_i^T is a number of topic words and w_i number of all words in i^{th} sentence
  - **Upscale** when sentence has many topic words
  - **Downscale** long sentences

Implementation in the model

Modifications

- Articles filtering - minimal topic words frequency
- Weights of words
  - **TF**
  - Modified **Dunning** measure
    - $log(D_i + 1)$ when $p_1 \geq p_0$
    - 0 when $p_1 < p_0$
  - Cosine **similarity** between word and topic embeddings
- Sentences' **embeddings**
- Ranking scaling - $F_i = \frac{log(w_i^T)}{log(w_i)}$
  - where F_i is a scaling factor, w_i^T is a number of topic words and w_i number of all words in i^{th} sentence
  - **Upscale** when sentence has many topic words
  - **Downscale** long sentences
- Non-duplicated sentences ($cos(\theta) > 0.5$)

Example

Summary

- Two days later, Mr Musk wrote a series of tweets including one describing Mr Unsworth as a **”pedo guy”**.
- Mr Unsworth’s legal team have described Mr Musk’s now-deleted **tweet** as ”vile and false” and are seeking unspecified punitive damages.
- On Thursday, Mr Unsworth told the **court** that Mr Musk’s tweet had left him feeling ”humiliated”.

GitHub

jkubajek/News_Selector

Bibliography i

E. Altszyler, M. Sigman, and D. F. Slezak.
Comparative study of LSA vs word2vec embeddings in small corpora: a case study in dreams database.
CoRR, abs/1610.01520, 2016.

T. Dunning.
Accurate methods for the statistics of surprise and coincidence.
COMPUTATIONAL LINGUISTICS, 19(1):61–74, 1993.

G. Erkan and D. R. Radev.
Lexrank: Graph-based lexical centrality as salience in text summarization.
Journal of artificial intelligence research, 22:457–479, 2004.

Bibliography ii

P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer.
Generating wikipedia by summarizing long sequences.
arXiv preprint arXiv:1801.10198, 2018.

R. Mihalcea and P. Tarau.
Textrank: Bringing order into text.
In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

L. Page, S. Brin, R. Motwani, and T. Winograd.
The pagerank citation ranking: Bringing order to the web.
Technical report, Stanford InfoLab, 1999.

G. Rossiello, P. Basile, and G. Semeraro.
Centroid-based text summarization through compositionality of word embeddings.
In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 12–21, 2017.