

Advanced Text Mining topics

MICHALIS VAZIRGIANNIS

October 2019

Outline

- **Topic modeling**
- Extractive summarization
- Graph based text Categorization

Topic Modeling

- Flux of information: Wikipedia articles, blogs, Flickr images, astronomical survey data, social networking activity
- Need algorithms *to organize, search and understand* this information.

Topic modeling

- aims at discovering the theme(s) of documents
- Aims at analyzing large quantities of unlabeled data.

Topic: probability distribution over a collection of words

Topic model

- statistical relationship between a group of *observed* and *latent* (unknown) random variables
- probabilistic procedure to generate the topics—a generative model.
- provides a “thematic summary” of a collection of documents (words’ distribution across topics)

Probabilistic LSA

- main goal: model co- occurrence information under a probabilistic framework to discover the underlying semantic structure of the data (Hofmann,1999)
- initially used for text-based applications (indexing, retrieval, clustering);
- spread in other fields: such as computer vision or audio processing.
- Goal of pLSA: use co-occurrence matrix to extract the “topics” and explain the documents as a mixture of them.

[Probabilistic Latent Semantic Analysis, Thomas Hofmann, in proceedings UAI 1999]

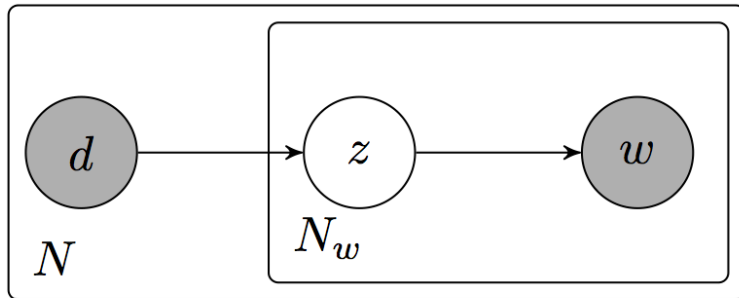
PLSA

Documents: $d \in D = \{d_1, \dots, d_N\}$ — observed variables, $|D| = N$

Words: $w \in W = \{w_1, \dots, w_M\}$ — observed variables, $|W| = M$

Topics: $z \in Z = \{z_1, \dots, z_K\}$ —latent (or hidden) variables.

$|Z| = K$, has to be specified a priori.



- graphical model representation.
- generative process for each of the N documents.
- N_w : number of words in document d .
- Each word w has associated a latent topic z from which is generated.
- Shaded circles: observed variables,

PLSA – Generative process

- select a document d with probability $P(d)$.
- for each word $w_i, i \in \{1, \dots, N\}$ in document d_n :
 - Select a topic z_i from a multinomial conditioned as $P(z | d_n)$.
 - Select a word w_i from a multinomial conditioned as $P(w | z_i)$.

Assuming

- bag-of-words model the joint distribution of the observed data factorize as a product:

$$P(\mathcal{D}, \mathcal{W}) = \prod_{(d,w)} P(d, w).$$

- Conditional independence

$$P(w|d) = \sum_{z \in \mathcal{Z}} P(w|z)P(z|d)$$

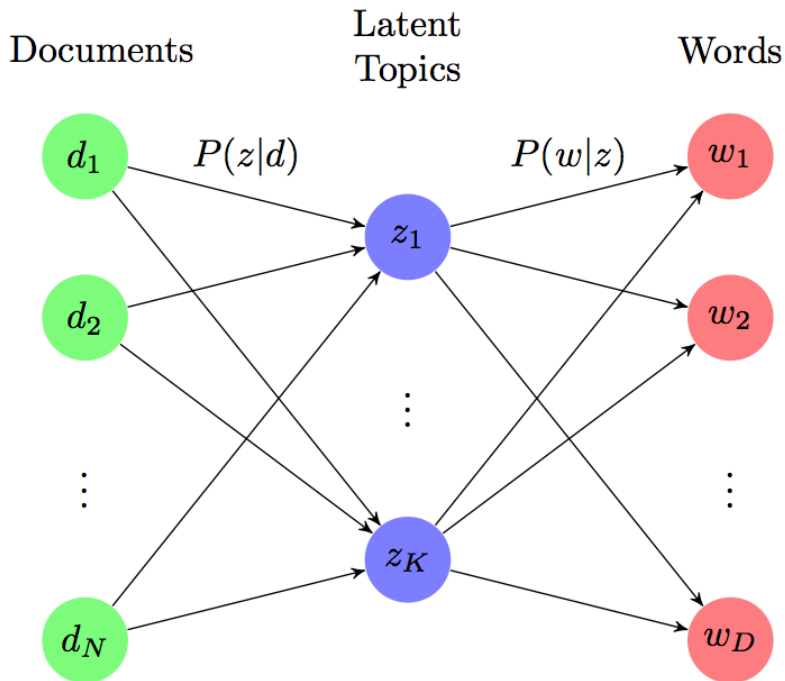
$$P(w, d) = \sum_{z \in \mathcal{Z}} P(z)P(d|z)P(w|z).$$

PLSA – mixture model

$$P(w|d) = \sum_{z \in \mathcal{Z}} P(w|z)P(z|d)$$

The general structure of pLSA model.

- intermediate layer of latent topics links documents to words
- each document is a mixture of topics weighted by the probability $P(z|d)$
- each word expresses a topic with probability $P(w|z)$.



$$L = \prod_{(d,w)} P(w|d) = \prod_{d \in \mathcal{D}} \prod_{w \in \mathcal{W}} P(w|d)^{n(d,w)}$$

$n(d, w)$ frequency of word w in d

PLSA – Log Likelihood Maximization

- Parameters can be estimated with Likelihood Maximization
- Find values maximizing predictive probability for observed word occurrences.
- predictive probability of pLSA mixture: $P(w/d)$, so the objective function is:

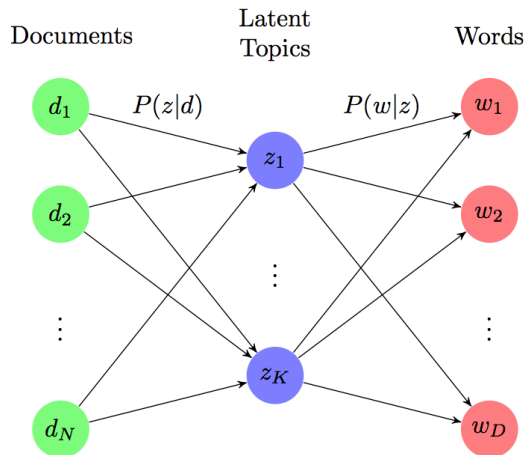
$$L = \prod_{(d,w)} P(w|d) = \prod_{d \in \mathcal{D}} \prod_{w \in \mathcal{W}} P(w|d)^{n(d,w)}$$

$n(d, w)$ frequency of word w in d

Can be solved with Expectation-Maximization (EM) algorithm for the log-likelihood:

$$\begin{aligned} \mathcal{L} = \log L = & \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(d, w) \\ & \cdot \log \sum_{z \in \mathcal{Z}} P(w|z)P(z|d). \end{aligned}$$

PLSA – as Matrix Decomposition



$$\begin{matrix} \xrightarrow{M} \\ N \end{matrix} \hat{A} = \begin{matrix} \text{green bar} \\ L \end{matrix} \times \begin{matrix} \text{blue diagonal} \\ U \end{matrix} \times \begin{matrix} \text{red bar} \\ R \end{matrix}$$

A: document-term matrix.

L: document probabilities $P(d|z)$.

U: diagonal matrix - prior probabilities of the topics $P(z)$.

R: word probability $P(w|z)$.

NMF for topic modelling

Explaining data by factorization

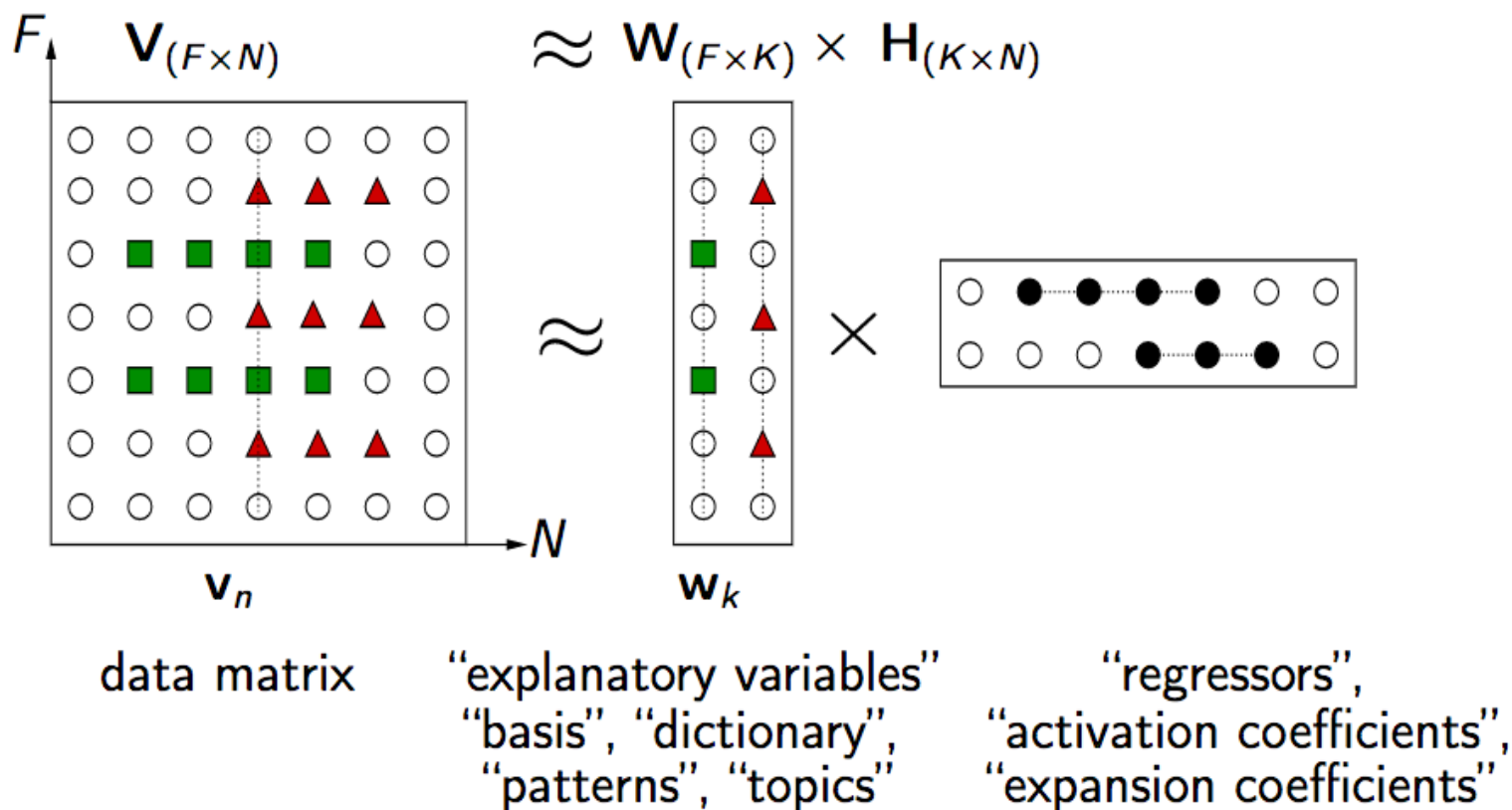


Illustration by C. Févotte

Non Negative Matrix factorization (NMF)

- - Data is often nonnegative by nature
 - - pixel intensities; occurrence counts; food or energy consumption; user scores; stock market values;
- - Interpretability of the results, optimal processing of nonnegative data may call for processing under Nonnegativity constraints
- .
- - Applying SVD results in factorized matrices with positive and negative elements may contradict the physical meaning of the result.
- - *Nonnegative matrix factorization (NMF)*
- find the reduced rank *nonnegative factors* to approximate a given nonnegative data matrix.

NMF model

- $V \simeq WH$
- $W = [w_{fk}], w_{fk} \geq 0$
- $H = [h_{kn}], h_{kn} \geq 0$
- $k \ll f, n$

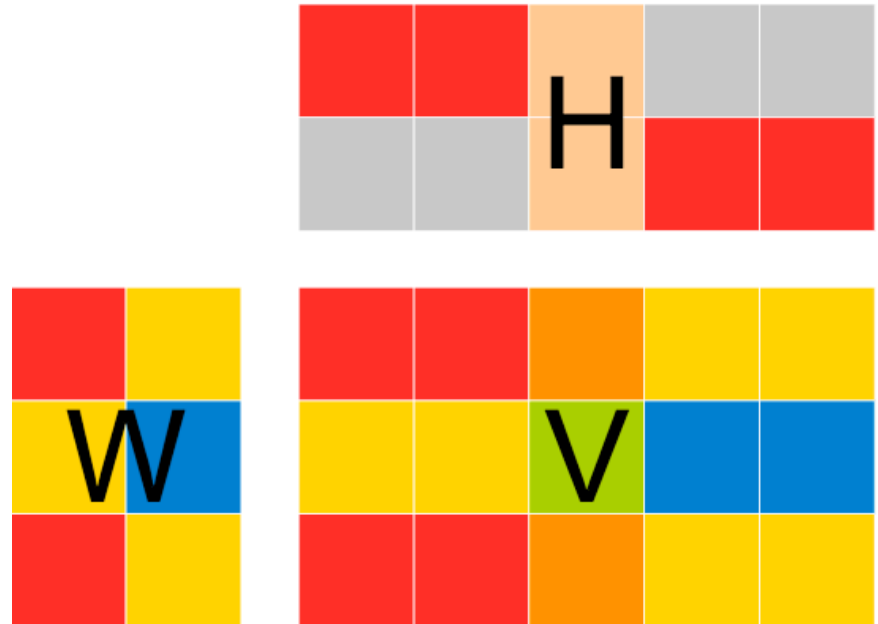


Illustration by N. Seichepine

NMF

- Assume X ($m \times n$) data matrix and $r \ll m, n$
- NMF aims to find non negative matrices

$$U \in R^{m \times r}, V \in R^{r \times n} : X \approx UV^T$$

- To find U, V , optimization problem:

$$\min_{(U, V)} ||X - UV^T||_2$$

- Alternative error function:

$$\min_{U, V} f(U, V) = \sum_i^m \sum_j^n (X_{ij} \log \frac{X_{ij}}{(UV^T)_{ij}} - X_{ij} + (UV^T)_{ij})$$

$$U_{ia} \geq 0, V_{jb} \geq 0, \forall i, a, b, j$$

Alternating Least squares

- 1. Suppose we know U , with V unknown.
- for each j we could minimize $\|X_{.j} - UV_{.j}^T\|_2$
 - find $V_{.j}$ that minimizes with $X_{.j}$ and U known.
 - Frobenius norm: sum of squares,
 - minimization is a least-squares problem, i.e. linear regression
 - “predicting” $X_{.j}$ from W .
- $$V_{.j} = (U^T U)^{-1} U^T X_{.j}$$
- - repeat for all columns $V_{.j}$
- 2. assume V , with U unknown $X^T = VU^T$
 - - Interchange roles of U , V in the above optimization
 - - Compute a row of U , repeat for all rows

Alternating Least squares

- Putting all this together
 - first choose initial guesses, random numbers, for U and V
 - alternate:
 - Compute U assuming V known
 - Compute V based on that new U
 - ...
- - may generate some negative values: simply truncate to 0

Other NMF Algorithms

- Multiplicative: updating solutions U and V

$$V_{bj}^{\top} \leftarrow V_{bj}^{\top} \frac{(U^{\top} X)_{bj}}{(U^{\top} U V^{\top})_{bj}} \quad U_{ia} \leftarrow U_{ia} \frac{(XV)_{ia}}{(UV^{\top} V)_{ia}}$$

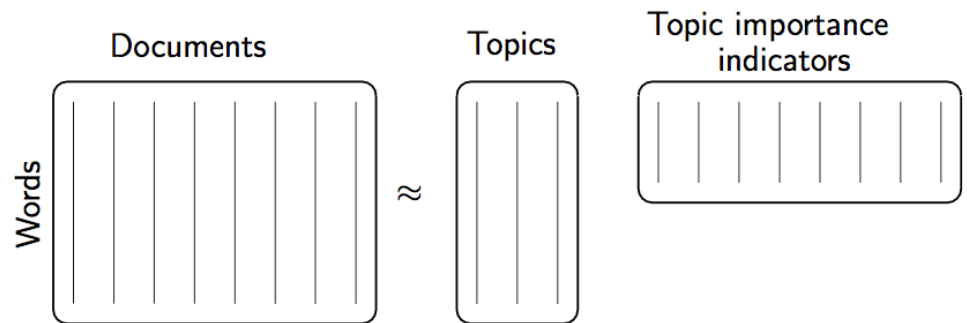
- Gradient descent algorithms

$$V_{bj}^{\top} \leftarrow V_{bj}^{\top} - \epsilon_V \frac{\partial f}{\partial V_{bj}^{\top}} \quad U_{ia} \leftarrow U_{ia} - \epsilon_U \frac{\partial f}{\partial U_{ia}}$$

- ϵ_V and ϵ_U are the step sizes.

NMF issues, applications

- Uniqueness and Convergence
- $U_{m \times r}$, r (rank) choice: via SVD...
- Applications
 - Topic detection
 - Source separation (music , speech)
 - Clustering
 - Recommendations



References

- “Latent Dirichlet Allocation: Towards a Deeper Understanding Colorado Reed January 2012
- D. Blei. Introduction to probabilistic topic models. Communications of the ACM, 2011.
- Probabilistic Topic Models, David M. Blei, Department of Computer Science Princeton University, September 2, 2012
- Probabilistic Latent Semantic Analysis, Dan Oneată
- Thomas Hofmann. Probabilistic latent semantic indexing. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR ’99, pages 50–57, New York, NY, USA, 1999. ACM

Outline

- Topic modeling
- **Extractive summarization**
- Text Categorization

Extractive summarization

Motivation

1. Keywords provide a snapshot of the ongoing topics
2. Can be used to improve participants experience
3. Utterances are often ill-formed
4. Speakers dilute information by pausing
5. Errors by the ASR
6. Traditional keyword extraction algorithms may not apply

Contribution

- A novel method to select representative words from automatic speech transcriptions in real-time.
- A thorough performance evaluation framework on the standard AMI and ICSI meeting corpora.

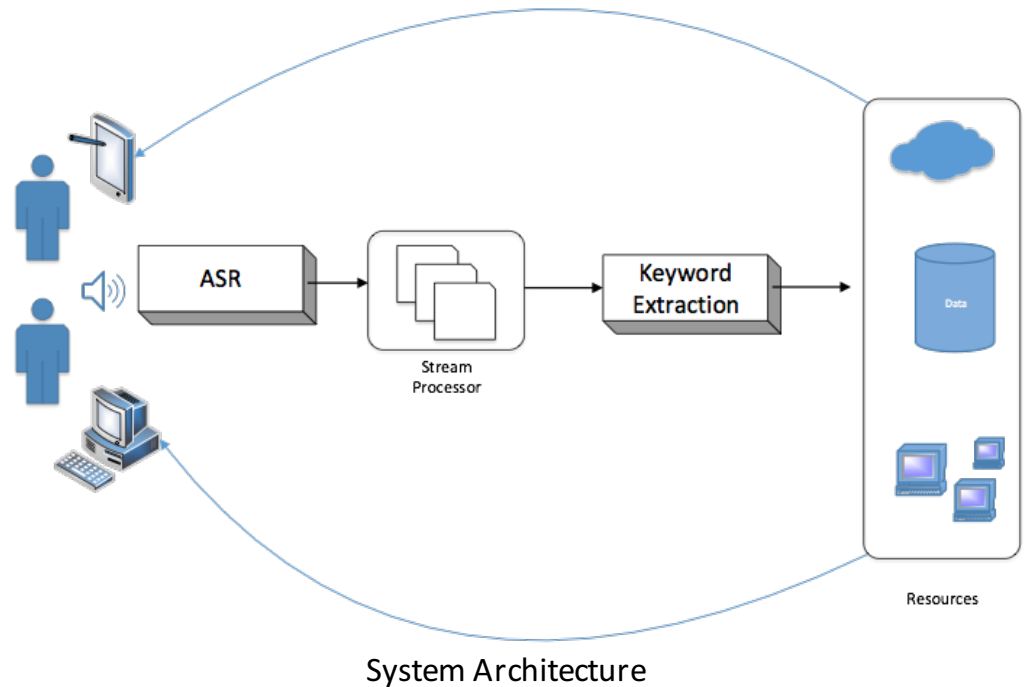
start	end	role	text
			Um well as the kick-off meeting for our our
55.415	60.35	PM	project i
60.35	64.16	PM	And um this is what we're
			this is what we're gonna be doing as an extra five
64.16	67.55	PM	minutes um
			um so of course we'll just a kind of make sure
67.55	72.79	PM	that we all know each other if i'm right
67.21	67.45	ME	Uh-huh
...

Example of utterances from the AMI corpus

System Architecture

Real Time Keyword Extraction

1. Stream Process: Retrieves the the ASR output
 - Time-intervals of 60 seconds
2. Keyword Extraction: Selects the most representative terms
3. Resource retrieval based on the extracted terms
4. Repeat steps

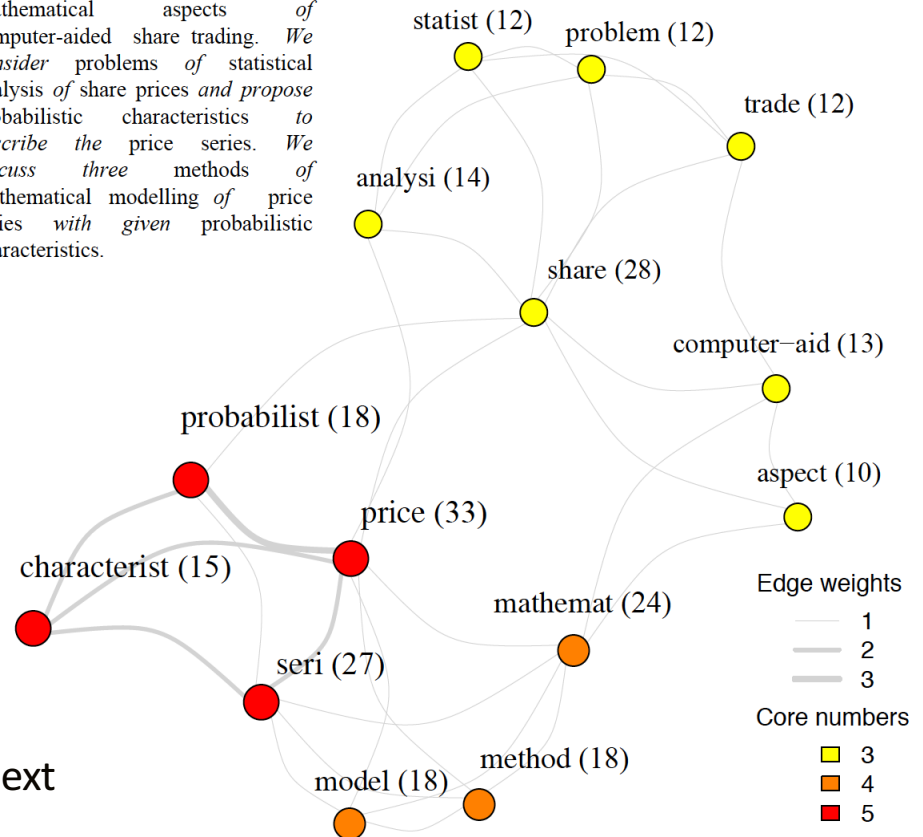


Representation

Graph-of-Words

1. Document as an *undirected, weighted* graph
2. Nodes are unique terms
3. Edge between two nodes if they co-occur within a fixed-size sliding window
4. Edge weights match co-occurrence counts

Mathematical aspects of computer-aided share trading. We consider problems of statistical analysis of share prices and propose probabilistic characteristics to describe the price series. We discuss three methods of mathematical modelling of price series with given probabilistic characteristics.



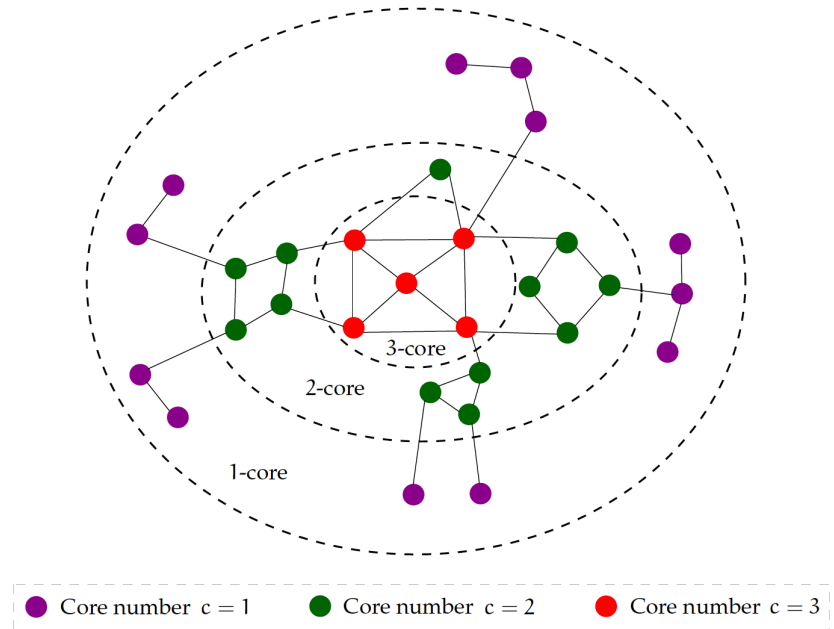
Pros

- powerful algorithms from graph theory to text
- Encodes the relative position of words

Graph-of-words example for a window of size 3.

Graph Decomposition –Core rank

- k-core of G is a maximal connected subgraph of G in which every vertex v has at least degree k
 - Degree of v is the sum of the weights of its incident edges
 - k-core decomposition: Hierarchy of nested subgraphs of decreasing size and increasing cohesiveness
 - Intuition: Cohesive components include keywords
-
- Coarse granularity of the k-core decomposition
 - Small Interval values worsen the problem
 - Many nodes may end up sharing the exact same core number
 - Work at the node level
 - Solves the ties



$$cr(v) = \sum_{u \in \mathcal{N}(v)} core(u)$$

Keyword extraction Quality function $f(S)$

$$f(S) = \sum_{v \in S} cr(v) - \lambda h(S)$$

- $h(S)$: the missing to complete graph links

$$h(S) = \binom{|S|}{2} - |E(S)|$$

- f satisfies the property of diminishing returns

$$f(A \cup v) - f(A) \geq f(B \cup v) - f(B)$$

for $A \subseteq B \subseteq V \setminus v$

- f is a submodular function

- Goal: maximize f

$$S^* = \arg \max_{S \subseteq V, \sum_{v \in S} c_v \leq B} f(S)$$

- Solution NP-Complete
- Greedy Algorithm approximation in polynomial time
- Performance Bound

$$(1 - 1/e) \approx 0.63$$

$$|S| \ll |V|$$

Evaluation

Datasets

- AMI -136 short meeting
- ICSI -56 long meetings

Evaluation Scenario 1:

- Streaming evaluation
- Ground truth: extractive summary
- For each time interval, keywords are compared to the part of the of the same time interval
- Metric: Cosine Similarity

$$\cos(\vec{d}_1, \vec{d}_2) = \frac{\vec{d}_1 \cdot \vec{d}_2}{\|\vec{d}_1\| \times \|\vec{d}_2\|}$$

Evaluation Scenario 2:

- Summarization evaluation
- Ground truth: abstractive summary
- Group together the keywords from all intervals
- Concise keyword-based summary of the meeting
- Metrics

- ROUGE

$$\text{ROUGE-N}(S) = \frac{\sum_{e \in R_i} \min(c_e(S), r_{e,i})}{\sum_{e \in R_i} r_{e,i}}$$

- WMD

$$\text{WMD}(\vec{d}_1, \vec{d}_2) = \min \sum_{i,j=1}^n T_{i,j} c_{i,j}$$



Word Mover's Distance as Evaluation Metric

$$\text{WMD}(\vec{d}_1, \vec{d}_2) = \min \sum_{i,j=1}^n T_{i,j} c_{i,j}$$

- minimum weighted cumulative cost needed for all words of d_1 to travel to d_2
- word-word traveling cost is computed as the distance in a highly-dimensional euclidean word embedding space
- Distributed representations of words encode syntactic and semantic regularities
- WMD can accurately measure the true dissimilarity between two documents

Baseline Methods

1. **Random**: terms are selected at random from the interval text. To reduce variance, we average results over 10 runs.
2. **Frequency**: this baseline uses the bag-of-words representation. The words with the greatest term-frequency (TF) are selected from the vocabulary.
3. **Degree**: the vertices associated with the highest degree centrality in the graph-of-words are returned.
4. **PageRank**: this method applies PageRank to the graph-of-words and returns the highest scoring nodes as keywords.
5. **RAKE**: The Rapid Automatic Keyword Extraction assigns scores $\text{deg}(v)/\text{freq}(v)$ to terms. Since utterances cannot be considered to be well-formed sentences, we use a sliding window like in our system.
6. **Oracle**: like the random baseline, but instead of drawing from the utterances, it draws from the human extractive summaries

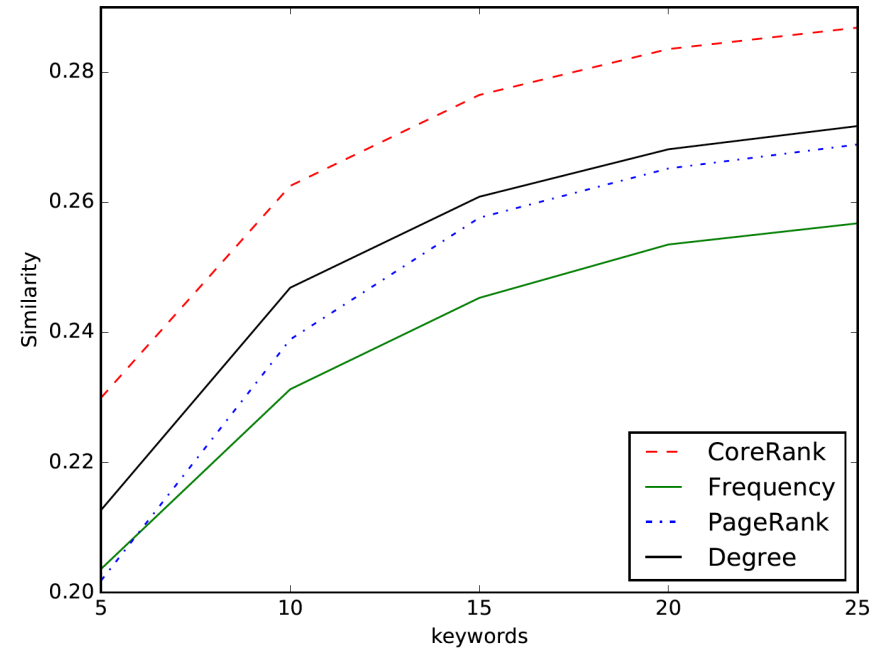
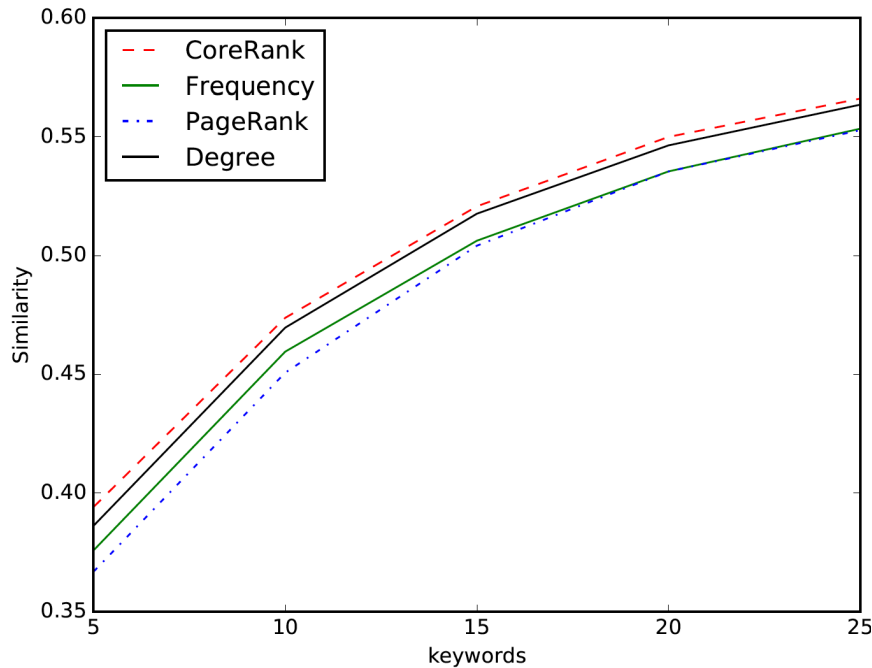
Results -Scenario 1

Method \ Dataset	AMI	ICSI
CoreRank	0.474*	0.259*
Frequency	0.460	0.231
PageRank	0.469	0.250
Random	0.365	0.190
Degree	0.470*	0.245
RAKE	0.384	0.196
Oracle	0.849	0.758

Real-time keyword extraction performance of
using **cosine similarity**.

*indicates statistical significance against the Frequency baseline of the same column.

Results -Scenario 1



Performance vs number of keywords using cosine similarity
for AMI corpus(left) and ICSI corpus(right)

Results -Scenario 1

Method \ Dataset	AMI	
	$\lambda = 0$	optimal λ
CoreRank	0.470	0.474
PageRank	0.466	0.469
Degree	0.467	0.470

Real-time keyword extraction performance with
 $\lambda = 0$ (left) and the optimal value of λ (right),
found using a small development set

Results -Scenario 2

<div>Dataset</div> <div>Method</div>	AMI		ICSI	
	Rouge	WMD	Rouge	WMD
CoreRank	0.237	1.6528	0.134	1.6986
Frequency	0.214	1.6610	0.121	1.7085
PageRank	0.219	1.6569	0.133	1.7011
Random	0.161	1.7610	0.077	1.7723
Degree	0.213	1.6570	0.130	1.7123
RAKE	0.195	1.7242	0.108	1.7050
Oracle	0.227	1.5816	0.136	1.0516

Real-time keyword extraction performance using
ROUGE (greater is better) and **WMD** (lower is better)

Outline.

Topic modelling

Advanced keyword extraction and summarization

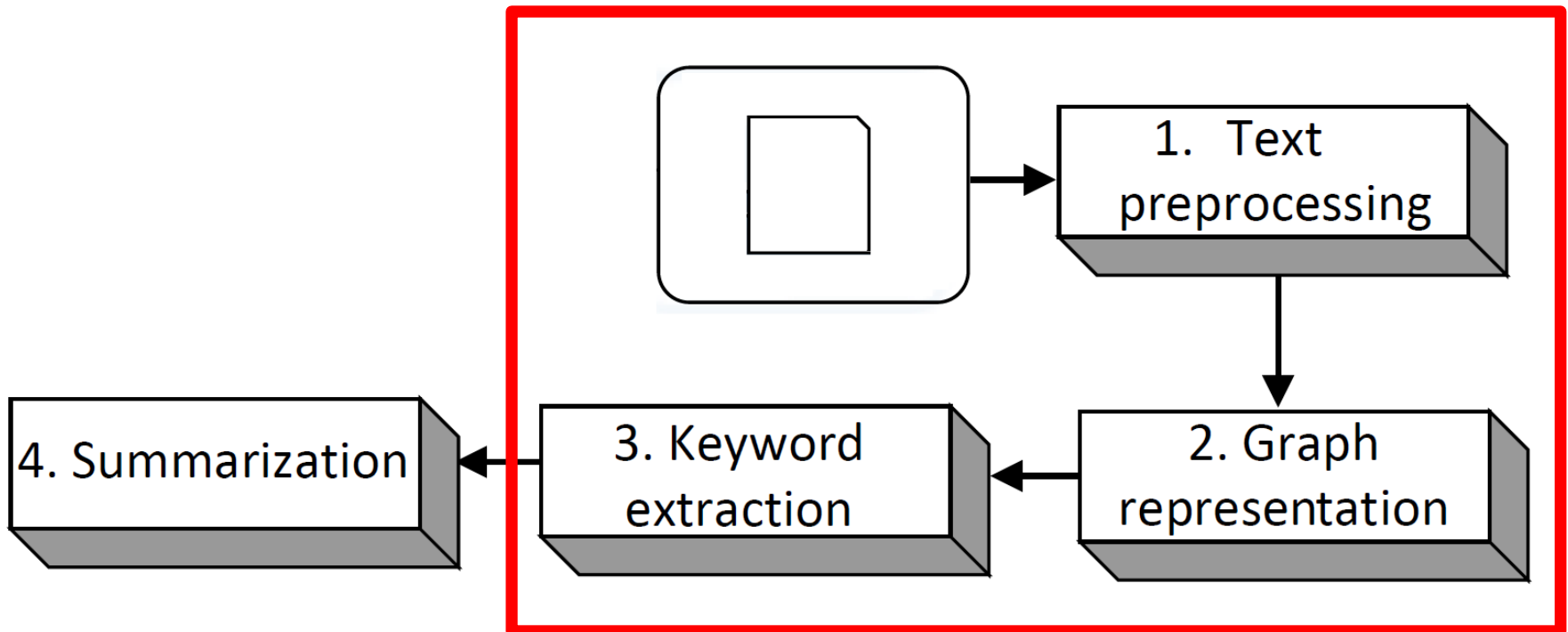
- Gow visualization & summarization - gowis – ACL 2016
- Heuristics based keyword extraction – EMNLP 2016
- Real time keyword extraction from online meetings (EACL 2017)
- Submodularity based Summarization (EMNLP - Workshop on New Frontiers in Summarization 2017, 48-58)

Advanced GoW topics

- Shortest-Path Graph Kernels for Document Similarity (submitted)
- Gaussian Document Representation from Word Embeddings

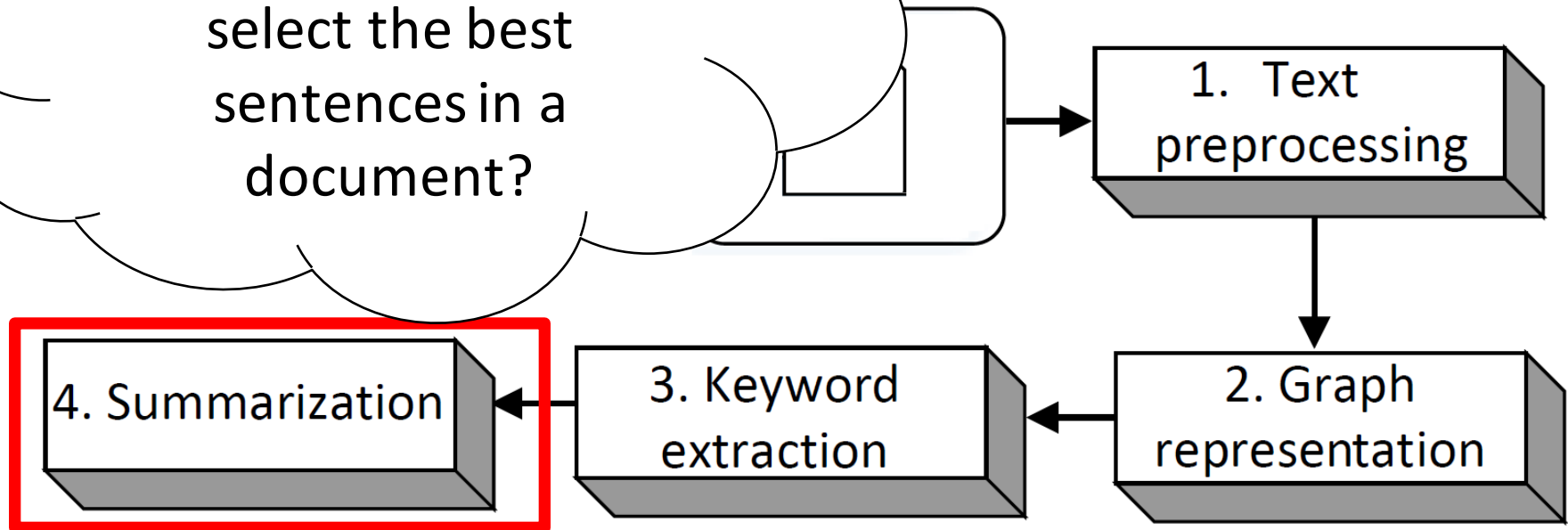
Extension to extractive document summarization

Same as before



Extension to extractive document summarization

How to use keywords
(and their scores) to
select the best
sentences in a
document?



Extension to extractive document summarization

Generating a summary in an **extractive** way is akin to selecting the best sentences in the document under a **budget constraint** (max number of words allowed).

-> **Combinatorial optimization** task:


$$\arg \max_{S \subseteq V} F(S) \mid \sum_{v \in S} c_v \leq B$$

- S is a given summary (a subset of the set of sentences V)
- F is the objective function to maximize (measuring summary quality)
- C_v is the cost of sentence v (number of words it contains)
- B is the budget (in words)


Extension to extractive document summarization

$$\arg \max_{S \subseteq V} F(S) \mid \sum_{v \in S} c_v \leq B$$

- Solving this task is **NP-complete**
- But, [7] showed that if F is **non-decreasing** and **submodular**, a greedy algorithm can approach the best solution with factor $(e - 1)/e$
- At each step, the algorithm selects the sentence v that **maximizes**:

objective function gain 

$$\frac{F(G \cup v) - F(G)}{c_v^r}$$

scaled cost 


- r is a tuning parameter

Extension to extractive document summarization

$$\arg \max_{S \subseteq V} F(S) \mid \sum_{v \in S} c_v \leq B$$

- The choice of F , the **summary quality** objective function, is what matters
- A good summary should cover all the important topics in the document, while not repeating itself:
 - > maximize **coverage**
 - > penalize **redundancy** (reward **diversity** to ensure monotonicity)

$$F(S) = L(S) + \lambda R(S) \quad \text{see [7]}$$

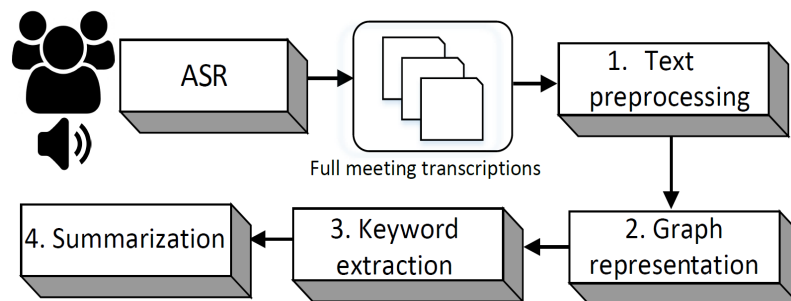

$$L(S) = \sum_{i \in S} n_i w_i \quad R(S) = N_{keywords \in S} / N_{keywords}$$

weighted sum of the keywords
contained in the summary

proportion of unique
keywords contained

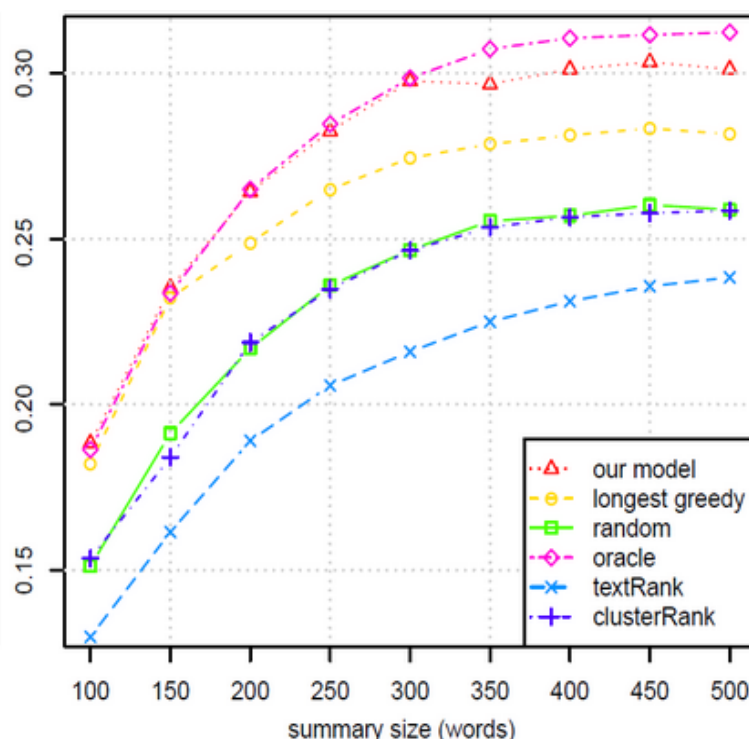
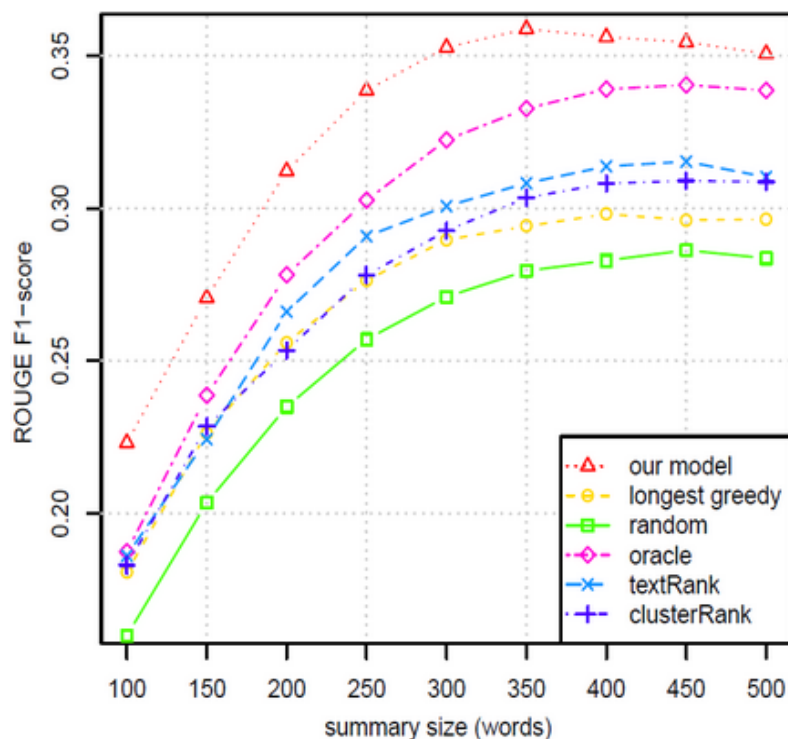
Extension to extractive document summarization

Tested for multiparty virtual meetings summarization:



AMI corpus

ICSI corpus



References

- [1] Seidman, S. B. (1983). Network structure and minimum degree. *Social networks*, 5(3), 269-287.
- [2] Cohen, J. (2008). Trusses: Cohesive subgraphs for social network analysis. *National Security Agency Technical Report*, 16.
- [3] Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., & Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature physics*, 6(11), 888-893.
- [4] Malliaros, F. D., Rossi, M. E. G., & Vazirgiannis, M. (2016). Locating influential nodes in complex networks. *Scientific reports*, 6.
- [5] Rousseau, F., & Vazirgiannis, M. (2015, March). Main core retention on graph-of-words for single-document keyword extraction. In *European Conference on Information Retrieval* (pp. 382-393). Springer International Publishing.
- [6] Mihalcea, R., & Tarau, P. (2004, July). TextRank: Bringing order into texts. Association for Computational Linguistics.
- [7] Lin, H., & Bilmes, J. (2010, June). Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 912-920). Association for Computational Linguistics.
- [8] Lin, H., & Bilmes, J. (2011, June). A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (pp. 510-520). Association for Computational Linguistics.

Outline

- Topic modeling
- Extractive summarization
- **Graph based text Categorization**

Text Categorization as a Graph Classification Problem [ACL2015]

- Single-label multi-class text categorization
- **Graph-of-words** representation of textual documents
- Mining of frequent **subgraphs as features** for classification
- Main core retention to reduce the graph's sizes
- **Long-distance n-grams** more discriminative than standard n-grams



Context

Applications of text classification are numerous:

- news filtering
- document organization
- spam detection
- opinion mining

Text documents classification compared to other domains:

- high number of features
- sparse feature vectors
- multi-class scenario
- skewed class distribution

Background

Text categorization [Sebastiani, 2002, Aggarwal and Zhai, 2012]

- Standard baseline: unsupervised n-gram feature mining + supervised linear SVM learning
- Common approach for spam detection: same with Naive Bayes

⇒ n-grams to take into account some **word order** and some **word dependence** as opposed to unigrams

⇒ word inversion? subset matching?

Background

- Graph classification
 - subgraphs as features
 - graph kernels [Vishwanathan et al., 2010]
 - Frequent subgraph feature mining
 - gSpan [Yan and Han, 2002]
 - FFSM [Huan et al., 2003]
 - Gaston [Nijssen and Kok, 2004]
 - expensive to mine all subgraphs, especially for “large” collections of “large” graphs
- ⇒ unsupervised discriminative feature selection?

Subgraph-of-words

- A subgraph of size n corresponds to a long-distance n -gram
⇒ takes into account **word inversion** and **subset matching**
- For instance, on the R8 dataset, {bank, base, rate} was a discriminative (top 5% SVM features) long-distance 3-gram for the category “interest”
 - “barclays **bank** cut its **base** lending **rate**”
 - “midland **bank** matches its **base rate**”
 - “**base rate** of natwest **bank** dropped”

!! patterns hard to capture with traditional n -gram bag-of-words.

Graph of Words Classification

Unsupervised feature mining and support selection

- gSpan mines the most frequent “subgraph-of-words” in the collection of graph-of-words
- subgraph frequency == long-distance n-gram document frequency
- minimum document frequency controlled via a **support** parameter
- the lower the support, the more features but the longer the mining, the feature vector generation and the learning
- \Rightarrow unsupervised support selection using the **elbow method** (inspired from selecting the number of clusters in k-means)

Multiclass Scenario

- Text categorization ==
multiple classes + skewed class distribution + single overall support value (local frequency)
- 100k features for majority classes vs. 100 features for minority ones
- \Rightarrow mining per class with same relative support value

Main core mining and n-gram feature selection

- Complexity to extract all features! \Rightarrow reduce the size of the graphs
 - Maintain word dependence and subset matching \Rightarrow keep the densest subgraphs
- \Rightarrow retain the main core of each graph-of-words
use gSpan to mine frequent subgraphs in main cores
- \Rightarrow extract n-gram features on remaining text (terms in main cores)

Experimental evaluation

Standard datasets:

- *WebKB*: 4 most frequent categories among labeled webpages from various CS departments – split into 2,803 for training and 1,396 for test [Cardoso-Cachopo, 2007].
- *R8*: 8 most frequent categories of Reuters-21578, a set of labeled news articles from the 1987 Reuters newswire – split into 5,485 for training and 2,189 for test [Debole and Sebastiani, 2005].
- *LingSpam*: 2,893 emails classified as spam or legitimate messages – split into 10 sets for 10-fold cross validation [Androutsopoulos et al., 2000].
- *Amazon*: 8,000 product reviews over four different sub-collections (books, DVDs, electronics and kitchen appliances) classified as positive or negative – split into 1,600 for training and 400 for test each [Blitzer et al., 2007].

Models

- 3 baseline models (n-gram features)
 - kNN (k=5)
 - Multinomial Naive Bayes (similar results with Bernoulli)
 - linear SVM
- 3 proposed approaches
 - gSpan + SVM (long-distance n-gram features)
 - MC + gSpan + SVM (long-distance n-gram features)
 - MC + SVM (n-gram features)

Evaluation metrics

- Micro-averaged F1-score (accuracy, overall effectiveness)
- Macro-averaged F1-score (weight each class uniformly)
- Statistical significance of improvement in accuracy over the n-gram SVM baseline assessed using the micro sign test ($p < 0.05$)
- For the Amazon dataset, we report the average of each metric over the four sub-collections

Effectiveness results (1/2)

<div>Dataset</div> <div>Method</div>	WebKB		R8	
	Accuracy	F1-score	Accuracy	F1-score
kNN (k=5)	0.679	0.617	0.894	0.705
NB (Multinomial)	0.866	0.861	0.934	0.839
linear SVM	0.889	0.871	0.947	0.858
gSpan + SVM	0.912*	0.882	0.955*	0.864
MC + gSpan + SVM	0.901*	0.871	0.949*	0.858
MC + SVM	0.872	0.863	0.937	0.849

Table: Test accuracy and macro-average F1-score. Bold font marks the best performance in a column. * indicates statistical significance at $p < 0.05$ using micro sign test with regards to the SVM baseline of the same column. gSpan mining support values are 1.6% (WebKB) and 7% (R8).

Effectiveness results (2/2)

Method \ Dataset	LingSpam		Amazon	
	Accuracy	F1-score	Accuracy	F1-score
kNN (k=5)	0.910	0.774	0.512	0.644
NB (Multinomial)	0.990	0.971	0.768	0.767
linear SVM	0.991	0.973	0.792	0.790
gSpan + SVM	0.991	0.972	0.798*	0.795
MC + gSpan + SVM	0.990	0.973	0.800*	0.798
MC + SVM	0.990	0.972	0.786	0.774

Table: Test accuracy and macro-average F1-score. Bold font marks the best performance in a column. * indicates statistical significance at $p < 0.05$ using micro sign test with regards to the SVM baseline of the same column. gSpan mining support values are 4% (LingSpam) and 0.5% (Amazon).

Dimension reduction – main core

Dataset	# n-grams before	# n-grams after	reduction
WebKB	1,849,848	735,447	60 %
R8	1,604,280	788,465	51 %
LingSpam	2,733,043	1,016,061	63 %
Amazon	583,457	376,664	35 %

Table: Total number of n-gram features vs. number of n-gram features present only in main cores along with the reduction of the dimension of the feature space on all four datasets.

Unsupervised support selection

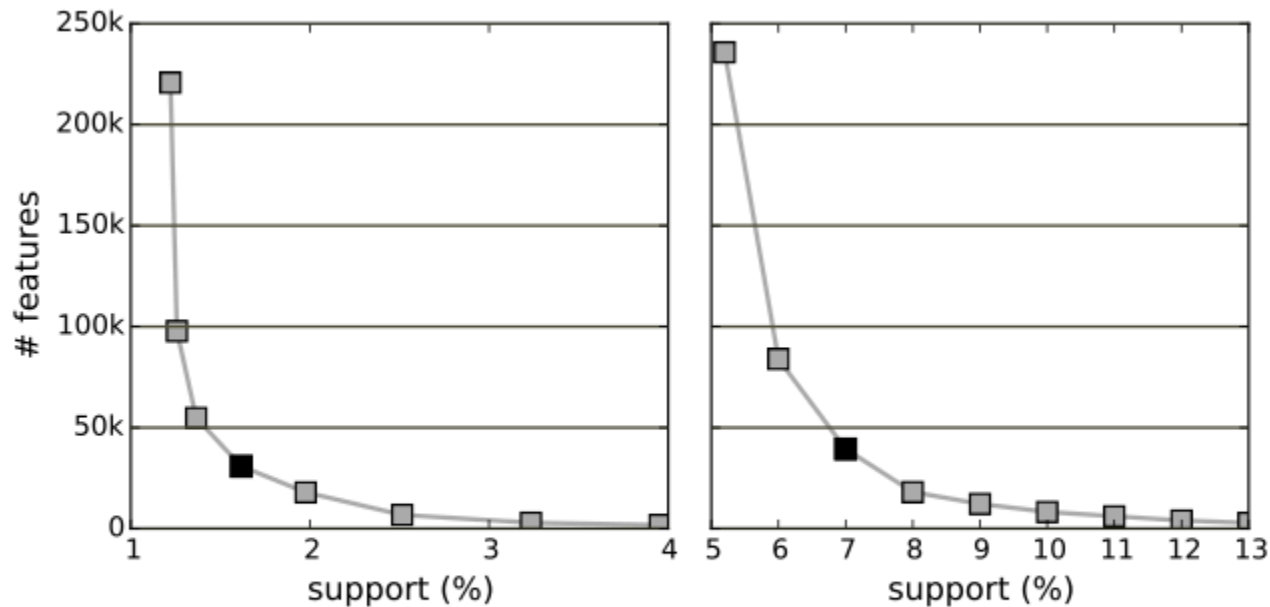


Figure: Number of subgraph features per support (%) on WebKB (left) and R8 (right) datasets. In black, the selected support chosen via the elbow method.

Distribution of mined n-grams

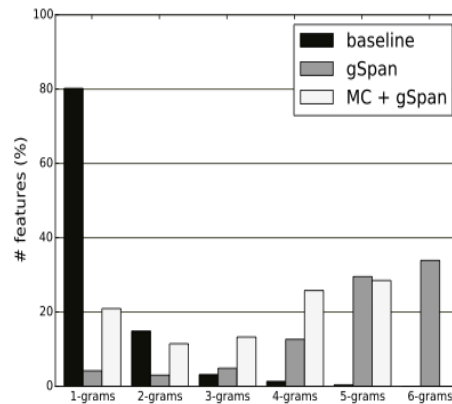


Figure: Distribution of n-grams (standard and long-distance ones) among all the features on WebKB dataset.

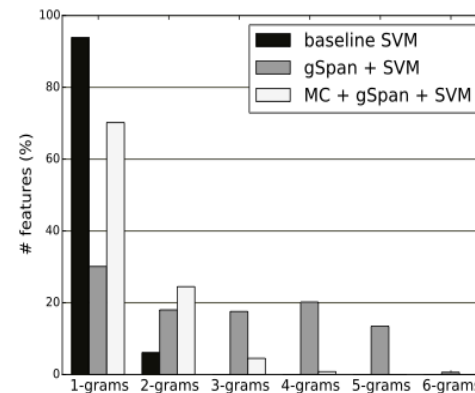


Figure: Distribution of n-grams (standard and long-distance ones) among the top 5% most discriminative features for SVM on WebKB dataset.

Contribution

- we explored a graph-of-words, to challenge the traditional bag-of-words for text classification.
- first trained a classifier using frequent sub- graphs as features for increased effectiveness.
- reduced each graph-of-words to its main core before mining the features for increased efficiency.
- reduced the total number of n-gram features considered in the baselines for little to no loss in prediction performances.

Shortest-Path Graph Kernels for Document Similarity (EMNLP 2017)

Bag Of Words

Traditionally, documents are represented as bag of words (BOW) vectors

- I entries correspond to terms
- I non-zero for terms appearing in the document

Example

- corpus vocabulary: {the, quick, brown, cat, fox, jumped, went, over, lazy, lion, dog}
- BOW representation of sentence: “the quick brown fox jumped over the lazy dog”

1	1	1	0	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---

- However, BOW representation disregards word order!!!

Bag of n-grams

n-gram: a contiguous sequence of n words

For example, the previous sentence: “the quick brown fox jumped over the lazy dog” contains the following tri-grams:

- | | |
|---------------------|--------------------|
| 1. the quick brown | 5. jumped over the |
| 2. quick brown fox | 6. over the lazy |
| 3. brown fox jumped | 7. the lazy dog |
| 4. fox jumped over | |

n-grams distinguish word order, however, they are *too strict*

- I unlikely that the same sequence of n word appears in independent documents

Graph of Words [Rousseau, 2013]

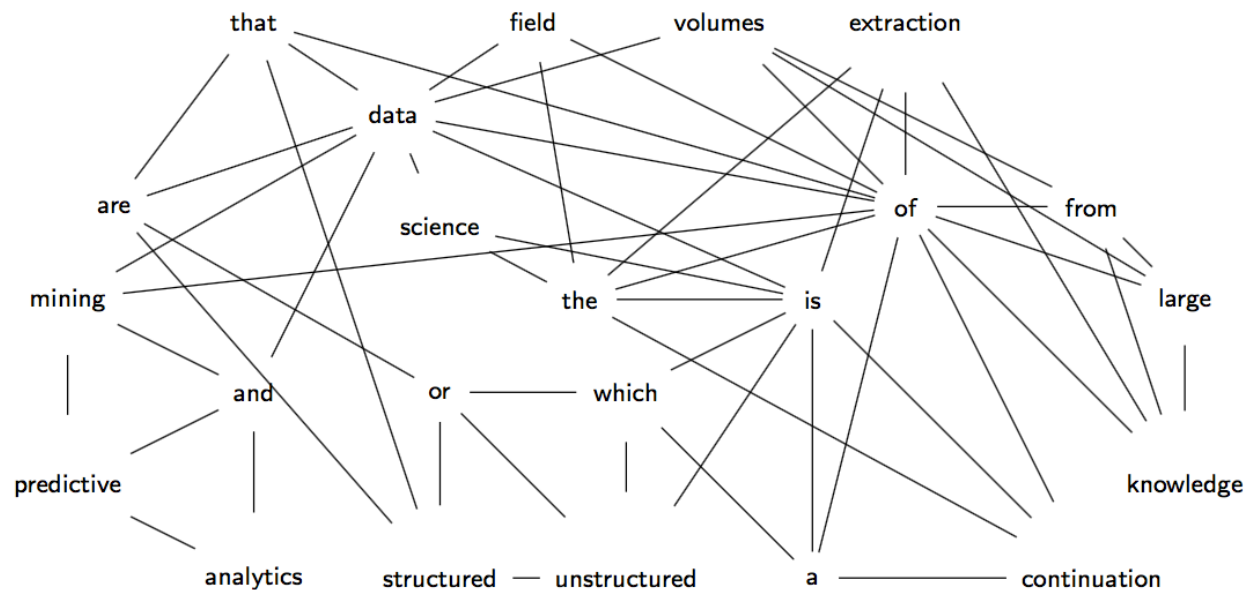
- Each document represented as a graph $G = (V, E)$ consisting of a set V of vertices and a set E of edges between them
- vertices \rightarrow unique terms (i.e. pre-processed words)
- edges \rightarrow co-occurrences within a fixed-size sliding window
no edge weight
- no edge direction

Graph representation more flexible than n-grams. It takes into account

- word inversion
- subset matching

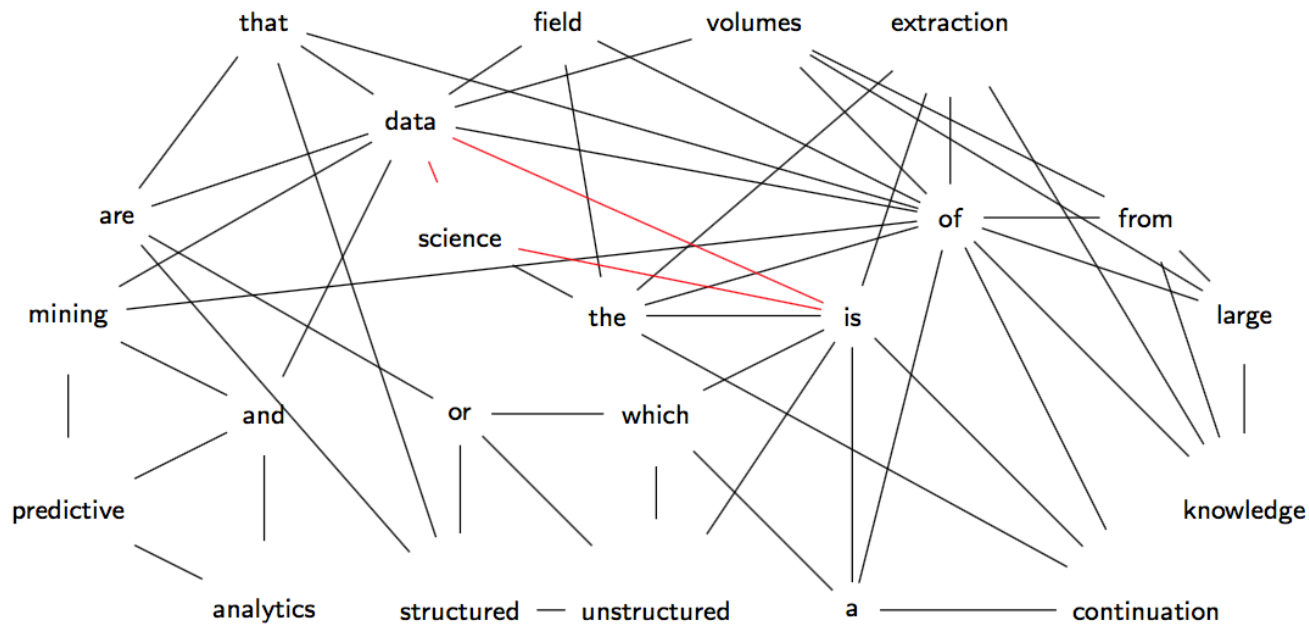
Graph of Words Example

Data Science is the extraction of knowledge from large volumes of data that are structured or unstructured which is a continuation of the field of data mining and predictive analytics



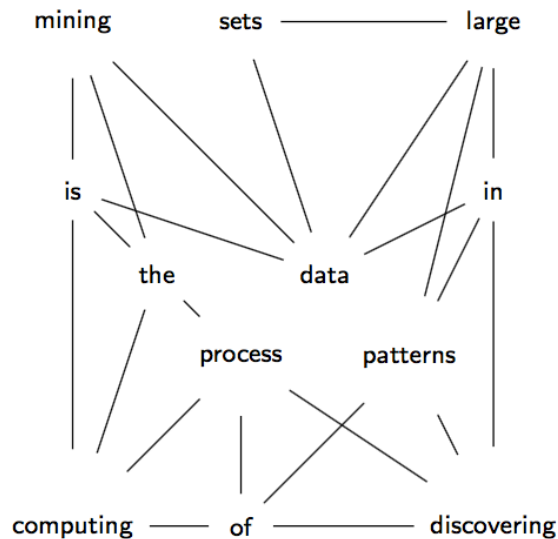
Graph of Words Example

Data Science is the extraction of knowledge from large volumes of data that are structured or unstructured which is a continuation of the field of data mining and predictive analytics

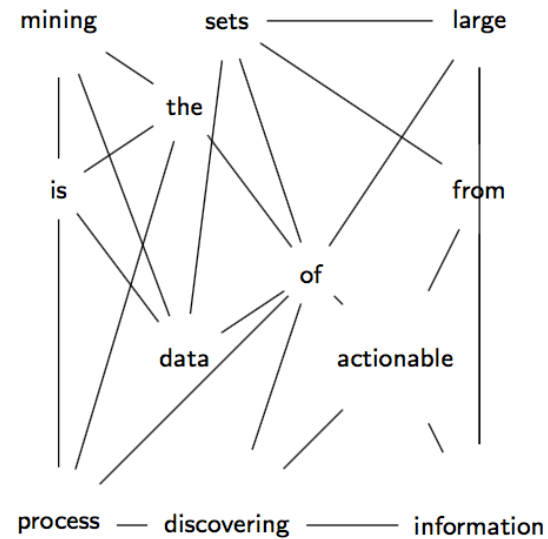


Document Similarity as Graph Comparison

Data mining is the computing process of discovering patterns in large data sets



Data mining is the process of discovering actionable information from large sets of data



Hence, **document** similarity problem → **graph** comparison problem

Applications of Graph Comparison

- Function prediction of chemical compounds
- Structural comparison and function prediction of protein structures
- Comparison of social networks
- Comparison of UML diagrams
- Document similarity

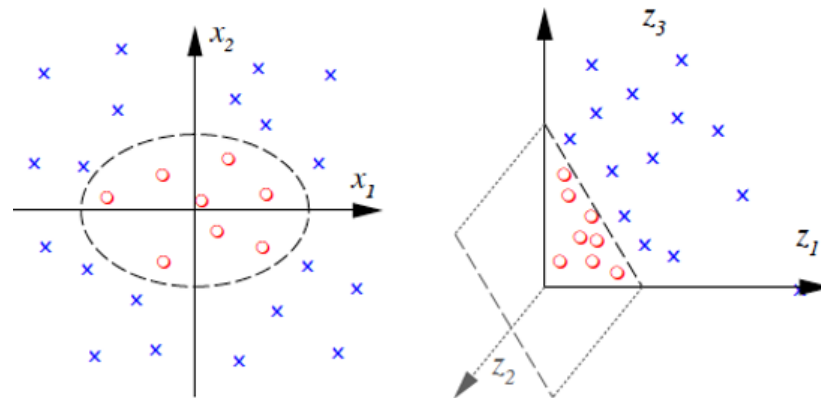
Approaches for Comparing Graphs

1. Graph isomorphism: find a mapping of the vertices of G_1 to the vertices of G_2 s.t. G_1 and G_2 are identical
 - No polynomial-time algorithm is known
 - Neither is it known to be NP-complete
2. Subgraph isomorphism: find if any subgraph of G_1 is isomorphic to a smaller graph G_2
 - NP-complete
3. Graph edit distance: count necessary operations to transform G_1 into G_2
 - Contains subgraph isomorphism check as one intermediate step
4. Graph kernels: compare substructures of graphs
 - Computable in polynomial time

Kernel Trick

The kernel trick avoids the explicit mapping that is needed to get linear learning algorithms to learn a nonlinear function or decision boundary

$$\Phi : R^2 \rightarrow R^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$



Kernels for Graphs

Graph representation of objects is advantageous compared to vectorial approaches :

- able to model relations between different parts of an object as well as the values of object properties
- The dimensionality of graphs is not fixed

Problem: How to compute kernels for structured data:

- Sequences
- Graphs

Custom Shortest Path Kernel

Based on the Shortest Path Kernel proposed by [Borgwardt and Kriegel, 2005]

Compares the length of shortest paths having the same source and sink labels in two graphs-of-words

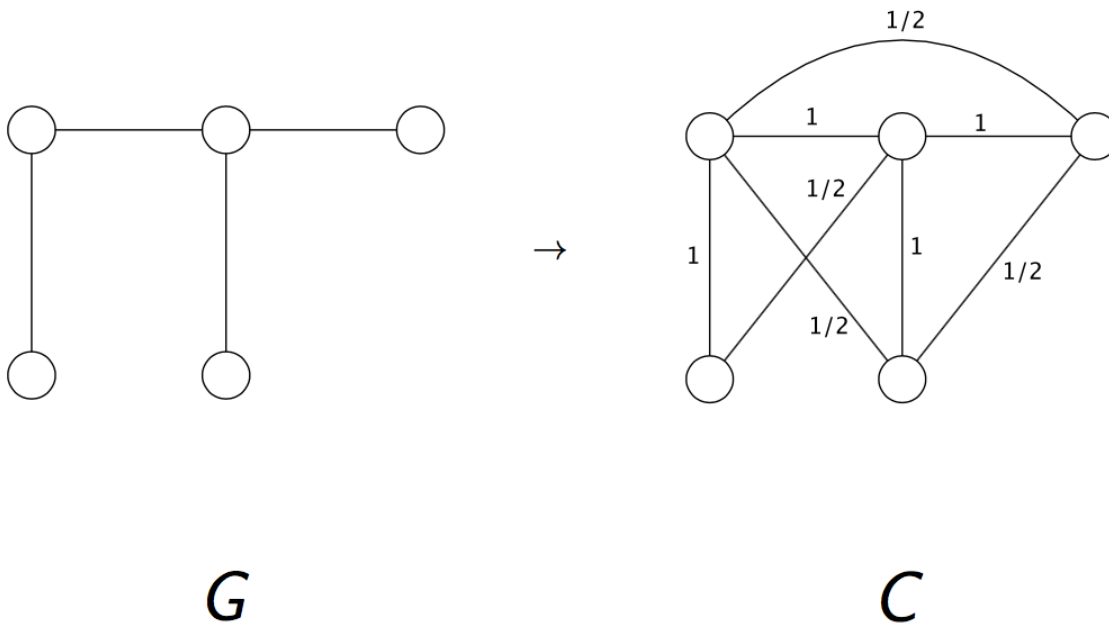
SP-transformation

Transforms the original graphs into shortest-paths graphs

- Compute the shortest-paths that are not greater than a variable d between all pairs of vertices of the input graph G using some algorithm (i. e. depth-first search)
- Create a shortest-path graph C which contains the same set of nodes as the input graph G
- All nodes which are connected by a path of length at most d in G are linked with an edge in C
- The label of an edge in C is set equal to $1/p$ where p is the length of the shortest path between its endpoints in G

Example

SP-transformation ($d = 2$)



Custom Shortest Path Kernel

Given the Floyd-transformed graphs $C_1 = (V_1, E_1)$ and $C_2 = (V_2, E_2)$ of G_1 and G_2 , the shortest path kernel is defined as:

$$k(G_1, G_2) = \frac{\sum_{v_1 \in V_1, v_2 \in V_2} k_{node}(v_1, v_2) + \sum_{e_1 \in E_1, e_2 \in E_2} k_{walk}^{(1)}(e_1, e_2)}{norm}$$

where k_{node} is a kernel for comparing two vertices, $k_{walk}^{(1)}$ a kernel on edge walks of length 1 and $norm$ a normalization factor. Specifically:

$$k_{node}(v_1, v_2) = \begin{cases} 1 & \text{if } \ell(v_1) = \ell(v_2), \\ 0 & \text{otherwise} \end{cases}$$

$$k_{walk}^{(1)}(e_1, e_2) = k_{node}(u_1, u_2) \times k_{edge}(e_1, e_2) \times k_{node}(v_1, v_2)$$

$$k_{edge}(e_1, e_2) = \begin{cases} \ell(e_1) \times \ell(e_2) & \text{if } e_1 \in E_1 \wedge e_2 \in E_2, \\ 0 & \text{otherwise} \end{cases}$$

Custom Shortest Path Kernel

Given the Floyd-transformed graphs $C_1 = (V_1, E_1)$ and $C_2 = (V_2, E_2)$ of G_1 and G_2 , the shortest path kernel is defined as:

$$k(G_1, G_2) = \frac{\sum_{v_1 \in V_1, v_2 \in V_2} k_{node}(v_1, v_2) + \sum_{e_1 \in E_1, e_2 \in E_2} k_{walk}^{(1)}(e_1, e_2)}{norm}$$

where k_{node} is a kernel for comparing two vertices, $k_{walk}^{(1)}$ a kernel on edge walks of length 1 and $norm$ a normalization factor. Specifically:

$$\mathbf{M}_1 = \mathbf{A}_1 + \mathbf{D}_1$$

$$\mathbf{M}_2 = \mathbf{A}_2 + \mathbf{D}_2$$

$$norm = \|\mathbf{M}_1\|_F \times \|\mathbf{M}_2\|_F$$

where $\mathbf{A}_1, \mathbf{A}_2$ are the adjacency matrices of the Floyd-transformed graphs, $\mathbf{D}_1, \mathbf{D}_2$ diagonal matrices with diagonal entries set to 1 if the corresponding term exists in the corresponding document and $\|\cdot\|_F$ is the Frobenius norm for matrices

Run Time Complexity

Standard kernel computation:

- All shortest paths from root: $\mathcal{O}(b^d)$ time (average branching factor b with breadth-first search)
- All shortest paths for n nodes: $\mathcal{O}(nb^d)$ time
- Compare all pairs of shortest paths from C_1 and C_2 : $\mathcal{O}(n^4)$
- However, since each node has a unique label, we have to consider n^2 pairs of edges
- Hence, total complexity: $\mathcal{O}(n^2 + nb^d)$

Special case for $d = 1$:

$$k(d_1, d_2) = \frac{\sum \mathbf{M}_1 \circ \mathbf{M}_2}{\|\mathbf{M}_1\|_F \times \|\mathbf{M}_2\|_F}$$

$\mathcal{O}(n + m)$ time in the worst case scenario

Evaluation

Document classification

- Goal: classify documents in a predefined set of categories
- Compute kernel matrix for all pairs of documents
- Support Vector Machine Model using the kernel matrices
- Compute Accuracy and F1 score

Link Detection

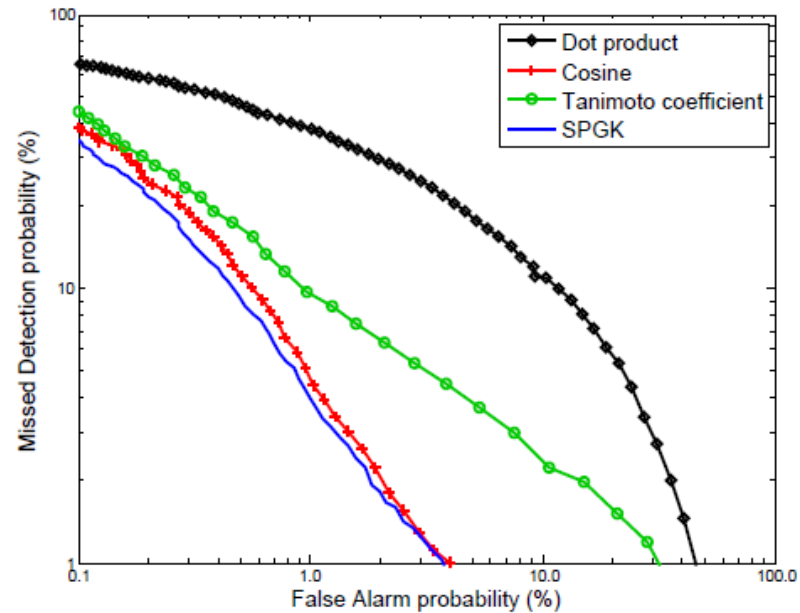
- Goal: find if two documents are linked
- Compute kernel distances for all pairs of documents
- Calculate Detection and Miss probabilities for every possible detection threshold (DET curve)

Classification results

Dataset		WebKB		News		Subjectivity		Amazon		Polarity	
Method		Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Dot product	$n = 1$	0.9026	0.8923	0.8110	0.7764	0.8992	0.8992	0.9188	0.9188	0.7627	0.7626
	$n = 2$	0.9047	0.8950	0.8091	0.7732	0.9101	0.9101	0.9200	0.9202	0.7746	0.7745
	$n = 3$	0.9026	0.8917	0.8072	0.7710	0.9090	0.9090	0.9181	0.9185	0.7741	0.7740
	$n = 4$	0.8940	0.8813	0.8031	0.7651	0.9039	0.9039	0.9131	0.9133	0.7719	0.7718
Cosine	$n = 1$	0.9248	0.9188	0.8117	0.7766	0.9003	0.9002	0.9400	0.9400	0.7670	0.7669
	$n = 2$	0.9305	0.9275	0.8149	0.7797	0.9094	0.9094	0.9413	0.9413	0.7756	0.7756
	$n = 3$	0.9298	0.9259	0.8097	0.7738	0.9099	0.9099	0.9419	0.9418	0.7765	0.7765
	$n = 4$	0.9248	0.9208	0.8076	0.7709	0.9076	0.9075	0.9413	0.9413	0.7753	0.7753
Tanimoto	$n = 1$	0.9062	0.8983	0.8155	0.7815	0.9094	0.9093	0.9225	0.9226	0.7749	0.7748
	$n = 2$	0.9040	0.8945	0.8075	0.7700	0.9061	0.9060	0.9181	0.9185	0.7735	0.7735
	$n = 3$	0.9241	0.9180	0.7980	0.7575	0.9021	0.9020	0.9344	0.9347	0.7648	0.7648
	$n = 4$	0.9176	0.9084	0.7899	0.7483	0.8953	0.8952	0.9300	0.9300	0.7586	0.7586
DCNN		0.8918	0.8799	0.7991	0.7615	0.9026	0.9026	0.9181	0.9181	0.7326	0.7326
CNN	static,rand	> 1 day		0.7757	0.7337	0.8716	0.8715	0.8881	0.8882	0.7150	0.7150
	non-static,rand	> 1 day		0.8113	0.7749	0.8961	0.8960	0.9356	0.9356	0.7654	0.7653
SPGK	$d = 1$	0.9327	0.9278	0.8104	0.7749	0.9148	0.9148	0.9400	0.9401	0.7776	0.7775
	$d = 2$	0.9370	0.9336	0.8089	0.7729	0.9146	0.9146	0.9413	0.9413	0.7789	0.7788
	$d = 3$	0.9291	0.9233	0.8078	0.7703	0.9137	0.9137	0.9444	0.9444	0.7761	0.7760
	$d = 4$	0.9291	0.9223	0.8097	0.7730	0.9118	0.9118	0.9463	0.9463	0.7780	0.7780

- On 4/5 datasets, SPGK outperforms the other three similarity measures and the NN architectures

Story Link Detection



References

- [1] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at International Conference on Learning Representations*, ICLR '13.
- [2] François Rousseau and Michalis Vazirgiannis. 2013. Graph-of-word and tw-idf: New approach to ad hoc ir. In *Proceedings of the 22nd ACM international conference on Information and knowledge management*, CIKM '13, pages 59–68.
- [3] Konstantinos Skianis, François Rousseau, Michalis Vazirgiannis. Regularizing Text Categorization with Clusters of Words. EMNLP 2016
- [4] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.
- [5] Vladimir Naumovich Vapnik. 1991. Principles of Risk Minimization for Learning Theory. In *Advances in Neural Information Processing Systems 4*, NIPS '91, pages 831–838.
- [6] Dani Yogatama and Noah A. Smith. 2014a. Linguistic structured sparsity in text categorization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL '14, pages 786–796.
- [7] Dani Yogatama and Noah A. Smith. 2014b. Making the most of bag of words: Sentence regularization with alternating direction method of multipliers. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *ICML '14*, pages 656–664.
- [8] Malliaros, F. D., Rossi, M. E. G., & Vazirgiannis, M. (2016). Locating influential nodes in complex networks. *Scientific reports*, 6.
- [9] Rousseau, F., & Vazirgiannis, M. (2015, March). Main core retention on graph-of-words for single-document keyword extraction. In *European Conference on Information Retrieval* (pp. 382-393). Springer International Publishing.
- [10] Mihalcea, R., & Tarau, P. (2004, July). TextRank: Bringing order into texts. Association for Computational Linguistics.
- [11] Lei Yuan, Jun Liu, and Jieping Ye. 2011. Efficient methods for overlapping group lasso. In *Advances in Neural Information Processing Systems 24*, NIPS '11, pages 352–360.

Relevant publications

- ① Rousseau, F. and M. Vazirgiannis (2013a). Composition of TF Normalizations: New Insights on Scoring Functions for Ad Hoc IR. In *Proceedings of ACM SIGIR '13*, pp. 917–920.
- ② Rousseau, F. and M. Vazirgiannis (2013b). Graph-of-word and TW-IDF: New Approach to Ad Hoc IR. In *Proceedings of the 22nd ACM CIKM '13*, pp. 59–68, [best paper mention award](#).
- ③ Rousseau, F. and M. Vazirgiannis (2015a). Main Core Retention on Graph-of-words for Single-Document Keyword Extraction. In *Proceedings of the 37th European Conference on Information Retrieval. ECIR '15*.
- ④ P. Meladianos, Y. Nikolentzos, F. Rousseau, Y. Stavrakas and M. Vazirgiannis (2015b) Degeneracy-based Real-Time Sub-Event Detection in Twitter Stream. Proceedings of the *9th AAAI International Conference on Web and Social Media (AAAI ICWSM '15)*.
- ⑤ Rousseau, F., Kiagias E. and M. Vazirgiannis, (2015c) Text Categorization as a Graph Classification Problem, in the proceedings of the *ACL 2015* conference
- ⑥ Jonghoon Kim F. Rousseau M. Vazirgiannis, “Convolutional Sentence Kernel from Word Embeddings for Short Text Categorization”, EMNLP 2015 conference
- ⑦ Konstantinos Skianis, François Rousseau, Michalis Vazirgiannis: Regularizing Text Categorization with Clusters of Words. EMNLP 2016: 1827-1837
- ⑧ Antoine J.-P. Tixier, Fragkiskos D. Malliaros, Michalis Vazirgiannis: A Graph Degeneracy-based Approach to Keyword Extraction. EMNLP 2016: 1860-1870
- ⑨ GoWvis: a web application for Graph-of-Words-based text visualization and summarization AJP Tixier, K Skianis, M Vazirgiannis ACL 2016, 151