Topic-based Multi-Document Summarization with Probabilistic Latent Semantic Analysis

Leonhard Hennig
DAI Labor, TU Berlin

Reviewed by -
Prashant Jalan
11523

What are they trying to do?

- Automatically (an unsupervised approach) create summaries that consist of the information most relevant with respect to a user's information need (query-focused summarization).

- DUC 2006 problem statement: Given a topic and a set of 25 relevant documents, the task is to synthesize a fluent, well-organized 250-word summary of the documents that answers the question(s) in the topic statement.

Related work

- Most existing approaches to summarization are sentence-based and extractive.
- Proposed approaches explore a variety of features, including document structure and term prominence, rhetorical structure, graph theoretic and semantic features.
- Topic information is usually encoded with term-based weighting strategies. Topic signatures are a well-known method of representing topic themes in multi document summarization.

What's new?

- Their system focuses on scoring sentences based on a representation of each sentence in the latent topic space provided by a trained **PLSA** model.
- Does not rely on the discriminative quality of the latent classes and does a redundancy check.
- Their approach utilizes narrative and other meta-information of the document cluster, too, to create query-focused summaries.

What's PLSA?

- In the paper 'Probabilistic Latent Semantic Analysis', **Thomas Hofmann** describes PLSA as a novel method for unsupervised learning based on a statistical latent class model. He argues that this approach is more principled than standard Latent Semantic Analysis, since it possesses a sound statistical foundation.
- He used tempered Expectation Maximization as a powerful fitting procedure.
- PLSA can be seen as a probabilistic analog of LSI (Latent Semantic Indexing).
- Latent semantic indexing (LSI) is an indexing and retrieval method that uses a mathematical technique called singular value decomposition (SVD) to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. LSI is based on the principle that words that are used in the same contexts tend to have similar meanings.

How do they do it?

After training a PLSA model on the term-sentence matrix of document clusters from recent summarization tasks, they represent each sentence as a distribution over latent topics. Using this representation, they combine query-focused and thematic sentence features into an overall sentence score. Sentences are ranked and selected for the summary according to this score, choosing a greedy approach for sentence selection and penalizing redundancy with a maximum marginal relevance method.

Preprocessing

- Given a corpus D of topic-related documents, they perform sentence splitting on each document using the NLTK toolkit2.
- Each sentence is represented as a bag-of-words w = (w1 , . . . , wm). During preprocessing, they remove stop words, and apply stemming using Porter's stemmer.
- They also discard all sentences which contain less than l_{min} = 5 or more than l_{max} = 20 content words, as these sentences are unlikely to be useful for a summary. We create a term-sentence matrix TS containing all sentences of the corpus, where each entry TS(i, j) is given by the frequency of term i in sentence j.
- They then train the PLSA model on the term-sentence matrix TS.

Applying the PLSA model

For each observation pair (d, w) the resulting likelihood expression is:

$$\begin{aligned} P(d,w) &= P(d)P(w|d), \text{ where} \\ P(w|d) &= \sum_{z \in Z} P(w|z)P(z|d). \end{aligned}$$

A document d and a word w are assumed to be conditionally independent given the unobserved topic z. Following the maximum likelihood principle, the mixing components and the mixing proportions are determined by the maximization of the likelihood function.

$$\mathcal{L} = \sum_{d \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(d,w) log P(d,w),$$

where n(d, w) denotes the term frequency, i.e. the number of times w occurred in d.

Sentence features

- Here they try to calculate the different sentence-level features such as sentence-narrative similarity r(S, N) or the sentence-document similarity r(S, D), or sentence-title similarity r(S, T).
- They use three similarity measures: The symmetric Kullback-Leibler (KL) divergence, the **Jensen-Shannon** (JS) divergence and the cosine similarity.

Sentence Scoring

$$score(s) = \sum_{P} w_p r_p^s,$$

- To compute the overall score of a sentence, they use the above formula where where w_p is a feature-specific weight.
- For their system, they trained the feature weights by initializing all weights to a default value of 1. They then optimized one feature weight at a time while keeping the others fixed.
- To deal with the problem of repetition of information, they model redundancy similar to the maximum marginal relevance framework. MMR is a greedy approach that iteratively selects the best-scoring sentence for the summary, and then updates sentence scores by computing a penalty based on the similarity of each sentence with the current summary.

Results

System	k	Rouge-1	Rouge-2	Rouge-SU4
PLSA-JS	192	**0.43283**	**0.09698**	**0.15568**
PYTHY	-	-	0.096	0.147
PLSA-COS	256	0.42444	0.09588	0.15409
peer 24	-	0.40980	0.09505	0.15464
PLSA-KL	256	0.42956	0.09465	0.15474
LSI	128	0.42155	0.08880	0.14938
Lead	-	0.30217	0.04947	0.09788

Table 1: *DUC-06: ROUGE recall scores for best number of latent topics k. PLSA-JS, -KL and -COS are system variants using the Jensen-Shannon-divergence, symmetric KL-divergence, and Cosine similarity respectively. Best LSI model based on a rank-k approximation with* $k = 128$.

System	k	Rouge-1	Rouge-2	Rouge-SU4
peer 15	-	0.44508	0.12448	0.17711
PLSA-JS	128	**0.45843**	**0.11675**	**0.17680**
PLSA-KL	224	0.45208	0.11662	0.17306
PLSA-COS	64	0.44329	0.11222	0.16679
LSI	256	0.44891	0.11022	0.16864
Lead	-	0.31250	0.06039	0.10507

Table 2: *DUC-07: ROUGE recall scores for best number of latent topics k. The PLSA-JS variant outperforms the best participating system on ROUGE-1, and ranks 5th for ROUGE-2.*

Conclusions

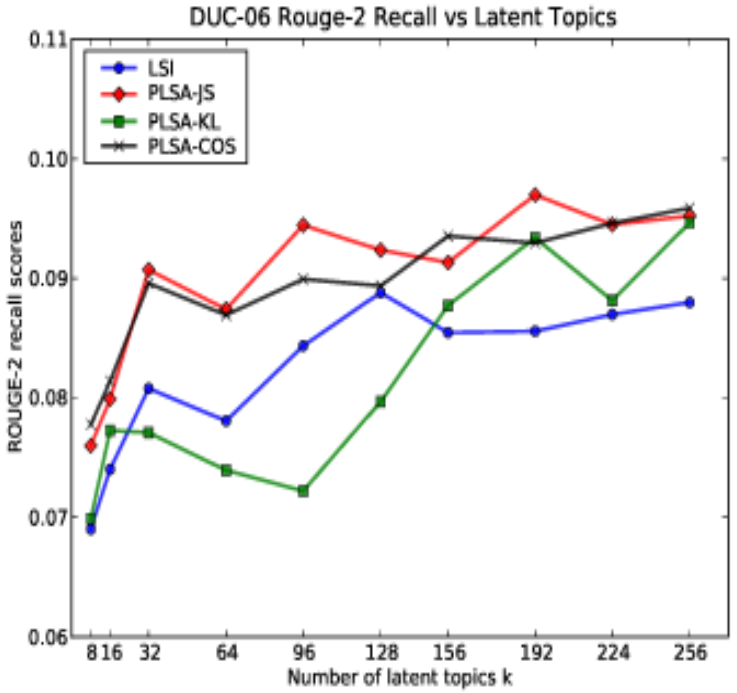
- PLSA model can better capture the sparse information contained in a sentence than a comparable LSI model.
- Using the Jensen-Shannon divergence resulted in the best ROUGE scores, as well as being very robust to changes of the number of latent classes.
- The performance differences of our system in terms of ROUGE-2 as compared to ROUGE-1 and ROUGE-SU4 suggests that the model could benefit from including n-gram co-occurrences.

Questions??

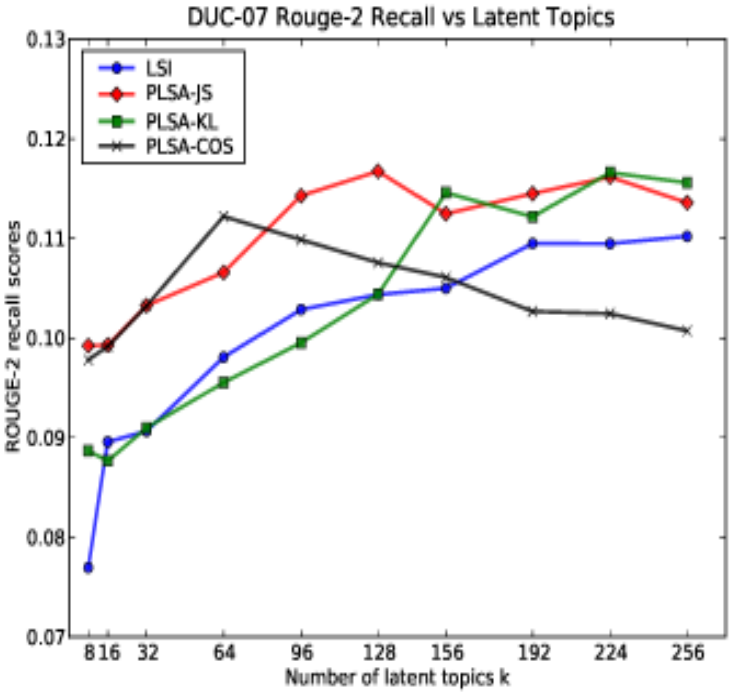
Evaluation

- Uses two data sets from recent summarization tasks: Multi-document summarization in DUC 2006 and in DUC 2007.
- For all evaluations, uses ROUGE metrics. ROUGE metrics are recall-oriented and based on n-gram overlap.
- They implemented two baseline systems, Lead and a system using LSI. The Lead system selects the lead sentences from the most recent news article in the document cluster as the summary. The LSI baseline computes the rank-k singular value decomposition of the term-sentence matrix.

Results



(a) DUC 2006



(b) DUC 2007

Fig. 1: *Summarization performance on DUC 2006 (a) and DUC 2007 data (b) in terms of ROUGE-2 recall for different numbers of latent topics. Also shown are the performance of system variants using different similarity measures and the LSI baseline. The system using Jensen-Shannon divergence with $k = 192$ latent classes outperforms existing approaches on DUC 2006 data.*

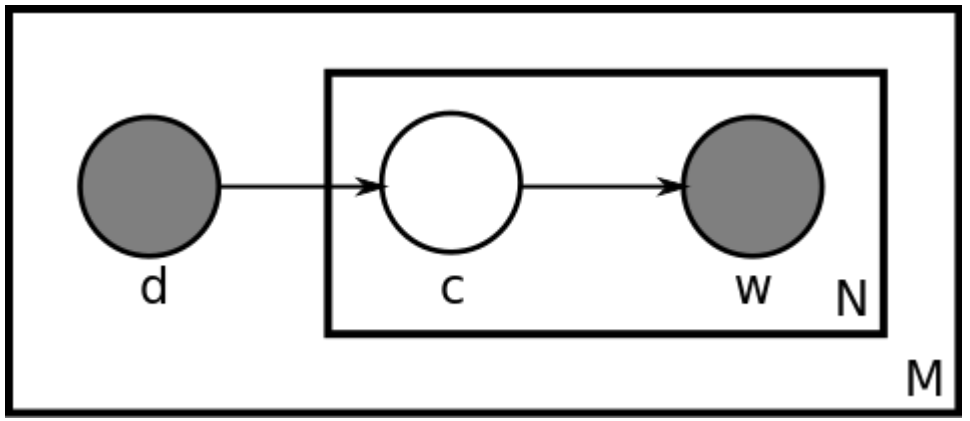
ROUGE?

ROUGE, or **Recall-Oriented Understudy for Gisting Evaluation** is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation.

The PLSA model

Considering observations in the form of co-occurrences (w,d) of words and documents, PLSA models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions:

$$P(w,d) = \sum_c P(c)P(d|c)P(w|c) = P(d)\sum_c P(c|d)P(w|c)$$



d is the document index variable, c is a word's topic drawn from the document's topic distribution, P(c|d), and w is a word drawn from the word distribution of this word's topic, P(w|c) . The d and w are observable variables, the topic c is a latent variable.

EM algorithm

EM is an iterative algorithm where each iteration consists of two steps, an expectation step where the posterior probabilities for the latent classes z are computed, and a maximization step where the conditional probabilities of the parameters given the posterior probabilities of the latent classes are updated. Alternating the expectation and maximization steps, one arrives at a converging point which describes a local maximum of the log likelihood. The output of the algorithm are the mixture components, as well as the mixing proportions over the components for each training document, i.e. the conditional probabilities P(w|z) and P (z|d).

Similarity measures

The symmetric KL-divergence is defined as follows:

$$\begin{aligned} KL(S,Q) &= D_{KL}(S||Q) + D_{KL}(Q||S) \\ &= \sum_I S(i) \log \frac{S(i)}{Q(i)} \\ &\quad + \sum_I Q(i) \log \frac{Q(i)}{S(i)}. \end{aligned} \tag{5}$$

To use the KL-divergence as a similarity measure, we scale divergence values to $[0, 1]$ and invert by subtracting from 1, hence

$$r_{KL} = 1 - KL(S,Q)_{scaled}. \tag{6}$$

The Jensen-Shannon divergence is a symmetrized and smoothed version of the KL-divergence, computing the KL-divergence of S, Q with respect to the average of the two input distributions. The JS-divergence based similarity r_{JS} is then defined as:

$$\begin{aligned} r_{JS}(S,Q) &= 1 - [D_{JS}(S||Q)] \\ &= 1 - \left[\frac{1}{2} D_{KL}(S||M) + \frac{1}{2} D_{KL}(Q||M)\right], \end{aligned} \tag{7}$$

where $M = 1/2(S + Q)$. Finally, the cosine similarity is defined as $r_{COS}(S,Q) = S^T Q$.