横浜国立大学 大学院 環境情報学府

情報メディア環境学専攻(前期)

言語情報処理原論(10)

Foundation of Natural Language Processing (10)

森 辰則

mori@forest.eis.ynu.ac.jp

今後の予定

□7/6(水) 講義 第11回目 (レポート出題予定)
□7/13(水)講義 第12回目 (最終回)
□7/20(水)休講 (大学院入試のため)
□7/27(水)休講 (試験週間のため)

文書情報の組織化(1)

多数の情報の中から利用者が要求する情報を捜し出す．これを効率良く行なうための各法

□選別
- ○情報検索
  - ▹情報が必要とされている時点での選別
  - ▹例えば，図書館での所蔵検索のように，すでにある文書群から知りたい事柄が記述されている文書を特定する
  - ▹検索要求は短期的
- ○情報フィルタリング
  - ▹情報が得られた時点での選別
  - ▹例えば，新聞記事の中から興味のある記事だけを拾い読みするように，時々刻々と配信される情報から必要なものを取り上げる．

文書情報の組織化(2)

□分類
- ○二つより多いグループに文書群を分類
  - ▹カテゴリ付与: あらかじめ与えられた分類体系に沿って文書を分類
  - ▹文書クラスタリング: 類似する文書をグループ化することによって分類．グループに対する適切な名前付け．

文書情報の組織化(3)

□抽出
- ○情報抽出，主題情報の抽出
  - ▹中心的な情報だけを抽出
  - ▹特定の記事カテゴリに対して抽出すべき情報がわかっていると仮定
  - ▹例：製品発表記事ならば，文書の中心的な情報は「製品名」「製品の価格」をぬき出せ

□要約
- ○文書の表す意味内容を非常に短いテキストで簡潔に表現する
- ○抽出した情報を文章の形で表現することに相当

情報検索

情報検索とは

□広義
- ○文書群において「知りたい」ことが記述されている文書を特定する
  - ▹利用者の未解決の問題を解決できる文書をみつけること

□狭義
- ○利用者の与えた検索質問文(query)に適合する(relevant)文書を見つける

利用者の要求(Information needs)

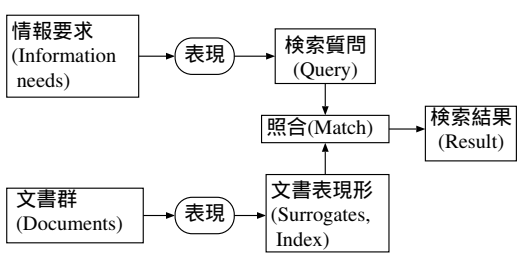
□直観的要求
- ○存在するがはっきりした情報要求の形にまではいたっていないもの

□意識された要求
- ○意識されており頭の中で記述できるもの

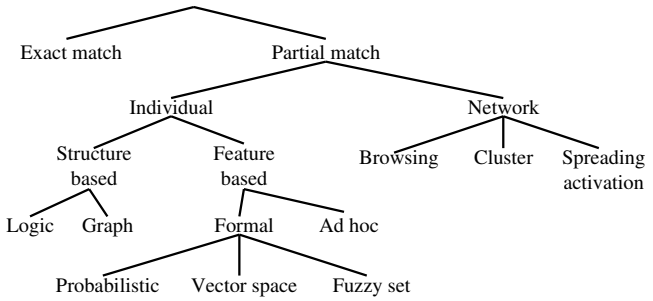
□具体化された要求
- ○要求の具体化された記述

□調整された要求
- ○情報システムへの質問

情報検索の枠組



情報検索の技術の分類



代表的な検索技術

□ブーリアンモデル

○伝統的なキーワード検索(Exact match)

□ベクトルモデル

○TFIDF法とベクトル空間法による検索

□確率モデル

○確率計算に基づく検索

準備: 索引語の抽出

□それぞれの文書に対してどのような「語」を索引語として採用するか

○「語」: Term = 語，語基，句，概念 など
○文書 → Termの並び

□不要語(stop word)

○どの文書にも現れ，文書検索の役に立たない語．
○これらは，前もって排除されることが多い．
○英語なら，冠詞,前置詞など．日本語なら，助詞，助動詞など．

ブーリアンモデル (boolean model) (1)

□一つの語，あるいは，複数の語を論理演算子で結合した論理式を検索質問として受け付ける．

□論理式が真となる文書(の集合)を結果として返す．

□主要論理演算子: ∧ (and), ∨ (or), ¬ (not) (これらと括弧'()')

○T (ある語): その語が索引語リストに含まれる文書で真，それ以外で偽
○Q1∧Q2: 論理式Q1ならびにQ2が共に真となる文書で真，それ以外で偽
○Q1∨Q2: 論理式Q1とQ2のうち少なくとも一方が真となる文書で真，それ以外で偽
○¬Q: 論理式Qが偽となる文書で真，それ以外で偽

□転置(インデクス)ファイルにより容易に適合文書を取り出せる

ブーリアンモデル (2)

欠点

□質問に部分一致する文書をとりこぼす
□検索した文書が順位づけされていない
□語の重要性を扱えない
□表現の語彙に影響されやすい

望ましい(最低限の)検索方式

「検索質問が与えられた時に語の重要性を考慮しつつ，文書の候補を順位をつけて表示する」

□語の重要性を表現できる

○索引語の抽出ならびに重みづけ

□検索された文書の順位づけ

○類似度を定義できるモデルで文書と質問を表現

ベクトルモデルと確率モデル

ベクトルモデル

TFIDF法とベクトル空間法による検索

□TFIDF法

○文書DB中の単語の分布に基づく単語の重要度計算法の一つ

□ベクトル空間法

○文書を多次元空間上のベクトルに対応させる手法

TF・IDF法

□ヒューリスティックな方法(多くの場合に有効だが，常にうまくいくという保証はない)

□文書DjにおけるTerm Tiの重要度w(i,j)

$$w(i,j) = TF(i,j) \cdot IDF(i)$$
$$IDF(i,j) = log_2(\frac{N}{DF(i)})(+1)$$

□TF(i,j): Term Frequency

○文書DjにおけるTerm Tiの出現回数
○包含性: ある文書においてその語が中心的話題であるか?

□DF(i): Document Frequency

○Term Ti を含む文書数

□IDF(i): Inverse Document Frequency(上式),Nは全文書数

○弁別性: 他の文書にはない事柄(語)であるか?

素性に基づく文書表現

□素性: 何らかの特徴量

□文書と質問を素性の組として表現

$e(f_1, f_2, \ldots, f_n)$

□素性の組の間の「類似度」を定義，その値により順位づけ

□代表的な方法

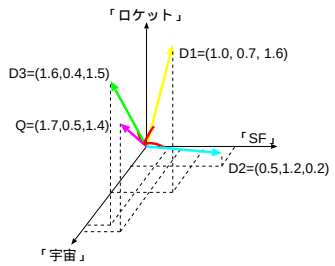
○ベクトル空間法(Vector Space Model)

□類似度は実数値となるために，「与えられた検索要求文」と「複数の文書」の間の類似度を計算し，順序づけすることができる．

ベクトル空間法(Vector Space Model) (1)

□文書と検索質問の両者を同一空間上のベクトルとして表現
□ベクトル間に類似度を定義し，類似文書の順位づけをする

□全文書の索引語Ti(i=1～t)に線形独立なt個のベクトルViを対応させる．



ベクトル空間法(2)

□このベクトル空間において文書Djを以下のように文書ベクトルで表現

$$D_j = \sum_i w(i,j) V_i$$

○w(i,j): 文書Djにおける Term Tiの重み

□検索質問についても同様に

$$Q_s = \sum_i q(i,s) V_i$$

○q(i,s): 検索質問Qsにおける Term Tiの重み

□例えば最も簡単なベクトルは

$$(w(1,j), w(2,j), ..., w(t,j))$$

ベクトル空間法における語の重み

文書ベクトル
□最も簡単なもの
○w(i,j) = 1 (文書jにTerm Tiが現れる場合)
○w(i,j) = 0 (それ以外)
□より高度なもの
○w(i,j) = TF(i,j)・IDF(i)

検索質問ベクトル
□最も簡単なもの
○q(i,s) = 1 (文書jにTerm Tiが現れる場合)
○q(i,s) = 0 (それ以外)
□より高度なもの

$$q(i,s) = (0.5 + 0.5 \cdot \frac{TF(i,s)}{\max_k \{TF(k,s)\}}) \cdot IDF(i)$$

ベクトル間の類似度(1)

cosine相関度(correlation)

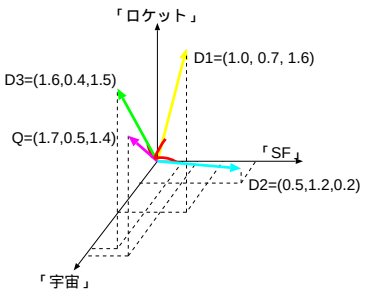
$$sim(D_j, Q_s) = \cos\theta_{j,s} = \frac{D_j \cdot Q_s}{|D_j||Q_s|}$$

□二つのベクトルのなす角のcosine値
○二つのベクトルが同じ方向を向いている
○cosの値が1に近いほど2つのベクトルが同じ方向を向いている
○二つのベクトルが異なる方向を向いている
○cosの値が0に近いほど2つのベクトルが異なる方向を向いている
□類似度がベクトルの長さは反映されていない
○語の強さの相対的な分布により類似度がきまる

ベクトル間の類似度(2)

例の場合の類似度

□D3とQ: 0.99
□D1とQ: 0.94
□D2とQ: 0.58



演習問題

文書1	文書2	文書3	文書4
. . 自動車 . . . . 鉄道 . . . . ロケット . .	ロケット . . . . 宇宙 . . . . ロケット .	. . 鉄道 . . . . 自動車 . .	. . 宇宙 . . . . ロケット . .

(宇宙, ロケット, 自動車, 鉄道)

	頻度ベクトル	TF・IDFベクトル
文書1	(0, 1, 1, 1)	(?,?,?,?)
文書2	(1, 2, 0, 0)	(?,?,?,?)
文書3	(0, 0, 1, 1)	(?,?,?,?)
文書4	(1, 1, 0, 0)	(?,?,?,?)

□tfidf法により文書ベクトルを求めよ．類似度の高い文書はどれ?
○ただし log2(4/3) = 0.4 とする．
○また，簡単のために各ベクトルの長さをすべて1.0と仮定する．
○(この仮定により，ある不都合が起きる．なぜか考えよ．)

確率モデル (1)

□仮定
○「ある文書dの検索質問qに対する関連度」
○＝「ある文書dj が検索質問qに関連している確率」
□求めるべき確率は，

$$P(R=1|D=d_j)$$

○確率変数R: 値は1もしくは0で，それぞれ，「qに関連している」「していない」
○確率変数D: 値は文書

確率モデル (2)

□文書の関連度の定式化
○確率P(R=1|D=dj)の順位を保存する関数LOR() (対数オッズ比, Log-odds ratio)とそれから導出される関連度関数sim()
▸ただし，O は "natural zero" (ゼロベクトル等)

$$\begin{aligned} LOR(d_j,q) &= \log \frac{P(R=1|D=d_j)}{P(R=0|D=d_j)} \\ &= \log \frac{P(D=d_j|R=1)P(R=1)}{P(D=d_j|R=0)P(R=0)} \\ &= \log \frac{P(D=d_j|R=1)}{P(D=d_j|R=0)} + \log \frac{P(R=1)}{P(R=0)} \\ sim(d_j,q) &= LOR(d_j,q) - LOR(O,q) \\ &= \log \frac{P(D=d_j|R=1)P(D=O|R=0)}{P(D=d_j|R=0)P(D=O|R=1)} \end{aligned}$$

確率モデル (3)

□文書の関連度から単語の重みへ

○文書djにおいて構成素(例えば単語i)間の独立性を仮定

$$\begin{aligned} sim(d_j,q) &= \log \frac{\prod_i P(T_i=t_i|R=1)\prod_i P(T_i=0|R=0)}{\prod_i P(T_i=t_i|R=0)\prod_i P(T_i=0|R=1)} \\ &= \sum_i \log \frac{P(T_i=t_i|R=1)P(T_i=0|R=0)}{P(T_i=t_i|R=0)P(T_i=0|R=1)} \\ &= \sum_i W(T_i=t_i) \end{aligned}$$

$$W(T_i=t_i) = \log \frac{P(T_i=t_i|R=1)P(T_i=0|R=0)}{P(T_i=t_i|R=0)P(T_i=0|R=1)}$$

○確率変数Ti : 第i番目の単語(単語i)に対する(ある文書における)統計量(例えば単語頻度など)

○W(Ti=ti) : 単語iについて，その統計量がtiであったときの単語の重要度

□確率P(Ti=ti|R=1)などをどのように推定するかが本質的な問題

確率モデル (4)

出現頻度を考慮しない場合

□次式となる

$$W(T_i=t_i) = \log \frac{p(1-q)}{q(1-p)}$$

○p: P(単語iが存在|R=1)

○q: P(単語iが存在|R=0)

□適当な確率推定により

$$W(T_i=t_i) \sim w^{(1)} = \log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)}$$

○N:全文書数， n:単語iが含まれる文書数，R:質問qに対して関連していることが既知である文書数，r:単語iが含まれ，かつ，質問qに関連していることが既知である文書数

○Robertson/Sparck Jones weight と呼ばれる

確率モデル (5)

出現頻度tf(i)を考慮する場合(1)

□準備: ポワソン分布 (Poisson distribution)

○生起頻度Nが低い事象について，単位時間内に平均でラムダ回発生する事象がちょうどk回(k=0,1,2,...)発生する確率

$$p(N=k,\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$$

□2つのポワソン分布(2-Poisson)に基づくモデル

○各語はある「エリート(elite)文書集合」に関連づけられている

▷エリート文書集合においては，その単語の文書内頻度tf(i)はポワソン分布に従う

▷残りの「非エリート」文書集合についても，ポワソン分布に従う．

$$W_i = \log \frac{(p'\lambda^{tf_i}e^{-\lambda}+(1-p')\mu^{tf_i}e^{-\mu})(q'e^{-\lambda}+(1-q')e^{-\mu})}{(q'\lambda^{tf_i}e^{-\lambda}+(1-q')\mu^{tf_i}e^{-\mu})(p'e^{-\lambda}+(1-p')e^{-\mu})}$$

○p': P(T_iに対するエリート文書集合|R=1)

○q': P(T_iに対するエリート文書集合|R=0)

□問題点:パラメタが4つもある．「エリート性」は隠れ変数．

確率モデル (6)

出現頻度tf(i)を考慮する場合(2)

□大幅な近似: 先の式の外形を模擬する．つまり，以下の特徴を持つ式で近似．

○a) tf(i)が0の時，値は0

○b) tf(i)について単調増加するが，c) ある最大値に漸近する

○d) Robertson/Sparck Jones weightに近い

□BM25 (BM = Best Match)

○Okapiシステムで採用されている

$$W_i = w^{(1)} \cdot \frac{(k_1+1)tf_i}{K+tf_i} \cdot \frac{(k_3+1)qtf_i}{k_3+qtf_i}$$

$$K = k_1((1-b)+b\frac{dl}{avdl})$$

○qtf(i): q内の単語iの頻度，dl:文書長，avdl:平均文書長

○k1,b,k3: パラメタ．Okapiシステムでは，k1=1.2, b=0.75,k3=7もしくは1000

類似度計算に基づく検索システム

検索の流れ(基本形)

□0) あらかじめ各文書に対する文書ベクトルを計算

□1) 検索質問を検索質問ベクトルに変換

□2) 検索質問ベクトルとすべての文書ベクトルとの類似度を計算する

□3) 全文書を類似度の大きい順に整列

□4) 上位M位までの文書を出力

すべての文書と類似度計算を行なうのは効率が悪い...

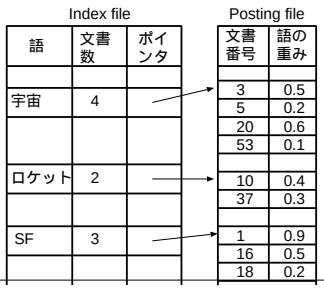
転置ファイル(Inverted file) による実装

転置ファイル

□辞書ファイルの一種

□語→ その語が出現している文書(あるいは出現位置)

□同時にその語に関する情報(語の重み等)も保存



転置ファイルによる実装 (cont.)

□0) あらかじめ転置ファイルを作成．ベクトル空間法の場合には，文書ベクトルの長さが1になるように語の重みを正規化．Inverted fileの各項目は以下の対応．

○Ti → w(i,n),...,w(i,m),...,w(i,l)

○ただし，w(i,m)は，文書mにおける語Tiの重みで，0でないもの

□1) 検索質問をベクトルQsに変換する．ベクトル空間法の場合には，Qsも正規化．

○Qs=(q(1,s),q(2,s),...,q(3,s))

○ただし，q(i,s)は，質問sにおける語Tiの重み

転置ファイルによる実装 (cont.)

□2) 0でないq(i,s)すべてについて以下を計算する．

○Inverted fileを調べ，Tiに対応するすべての w(i,m)について以下の計算をする．

○Sm ← Sm + w(i,m)*q(i,s)

○mの値はとびとびなので，Smの保存にはハッシュなどを利用する．

□3) Smに類似度が得られる．0でないものを整列する．

検索質問文に登場する語を含まない文書は全く計算しないので，文書数が多くなっても速度が低下しない．(辞書ファイルの作り方にも依存)

情報検索システムの性能評価

検索質問文に対して定義される，再現率と適合率が良く用いられる．

以下の定義において「適合文書」は「その検索質問文に適合する文書」

□再現率(recall) R

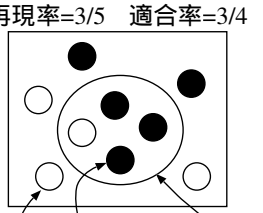
$$R = \frac{\text{検索された適合文書数}}{\text{全文書中の適合文書数}}$$

□適合率(precision) P

$$P = \frac{\text{検索された適合文書数}}{\text{検索された文書数}}$$

情報検索システムの性能評価(cont.)

再現率=3/5　適合率=3/4



適合文書　検索結果

適合しなかった文書

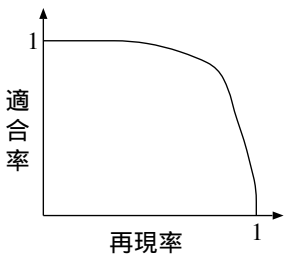
情報検索システムの性能評価(cont.)

□理想は

○再現率=1.0

○適合率=1.0

□しかし，実際には再現率と適合率の間にトレードオフの関係



検索効率の改善

□再現率が低いとすると，利用者はすべての適合文書のうち一部だけしか得ることができない

○検索条件を適応的に変化させる

▹検索要求の条件を利用者に緩めてもらう?

▹検索要求を変更してもらう(値の選択しなおし)

▹「適合性フィードバック」，「質問拡張」，「概念検索」

□適合率が低いとすると，利用者はかなりの数の無関係な文書を読まなければならない

○文書群の中から類似性を見つける

▹「係受け情報の利用」，「共起情報の利用」

適合性フィードバック

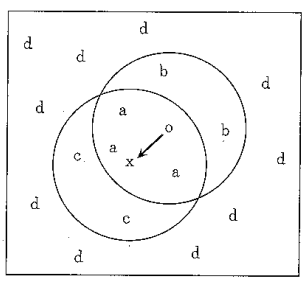
□Rocchio他，多数の研究者は，

□一度ではなく，数回の検索を繰り返す間に，徐々に結果を利用者の求めるものに近付けていく．

□検索が行なわれるたびに，利用者に検索された文書の評価をしてもらう(明示的/非明示的に)．

□検索質問ベクトルを「適合」した文書のほうに移動すれば，検索結果の中の「適合」文書が多くなるはず．

適合性フィードバック (cont.)



o：検索質問
a：検索された該当文書
b：検索された非該当文書
x：新しい検索質問
c：新たに検索された文書
d：その他の文書

図 2.8　関連フィードバックの概念図
先の検索結果に対するユーザーのフィードバック情報を元に，検索質問ベクトルを修正し，該当文書の中心により近付けることを行なう．

適合性フィードバック (cont.)

質問拡張 + 再重みづけ による方法(Standard Rocchio)

$$Q = Q_0 + \beta \sum_{i}^{n_1} \frac{R_i}{n_1} - \gamma \sum_{j}^{n_2} \frac{S_j}{n_2}$$

○Q0: 初めの質問に対応する検索質問ベクトル
○Ri: 適合文書Diの文書ベクトル
○n1: 適合文書の数
○Sj: 適合しない文書Djの文書ベクトル
○n2: 適合しない文書の数
○β, γ: パラメタ

効果的な方法

□適合文書のベクトルの成分 → 値が大きくなり強化される

□非適合文書のベクトルの成分 → 値が小さくなり薄まる

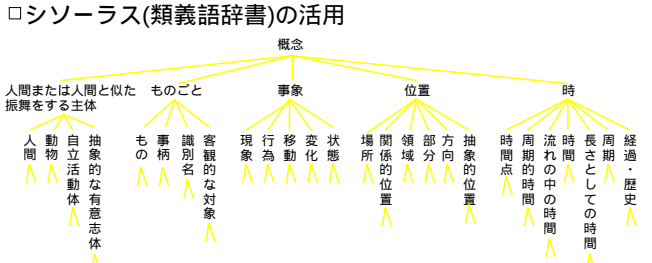
質問拡張(Query Expansion)

索引(辞書)にない言葉が検索質問に現れたら？

□無視してしまう方法では再現率が落ちる

□索引に現れる言葉のうち，質問中の語と「関連する」語を検索質問に加える

□シソーラス(類義語辞書)の活用



概念検索(Concept based IR)

□検索質問文における表層表現だけではなく，その「意味」(概念)をあつかう．

□(外部から与える)シソーラスによる質問拡張は概念検索の第一歩
- ○ポイントは語と語の間の類似性を導くこと

□シソーラス自動構築には，文書集合全体の情報を使い，語と語の間の類似度を求める
- ○「出現する文書(複数)が同じであるほど，類似度が高い」
  - ▹LSI
  - ▹CLARIT(Concept Base)
  - ▹Concept based Query Expansion
- ○「周りに現れる語の傾向が同じであるほど，類似度が高い」
  - ▹InfoMap

概念検索(Concept based IR) (cont.)

例:「出現する文書が同じであるほど，類似度が高い」

文書1	文書2	文書3	文書4	文書5
..宇宙..	..自動車..	..宇宙..	..鉄道..	..宇宙..
..ロケット..	..鉄道..	..ロケット..	..自動車..	..ロケット..
	..ロケット..			

語ベクトル	文書1	文書2	文書3	文書4	文書5
「宇宙」	1	0	1	0	1
「ロケット」	1	1	1	0	1
「自動車」	0	1	0	1	0
「鉄道」	0	1	0	1	0

類似度が高い組は 「宇宙」と「ロケット」
「自動車」と「鉄道」

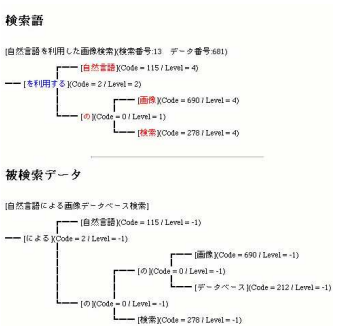
検索効率の改善(再掲)

□再現率が低いとすると，利用者はすべての適合文書のうち一部だけしか得ることができない
- ▹検索要求の条件を利用者に緩めてもらう?
- ▹検索要求を変更してもらう(類似語の追加など)
- ○検索条件を適応的に変化させる
  - ▹「適合性フィードバック」，「質問拡張」，「概念検索」

□適合率が低いとすると，利用者はかなりの数の無関係な文書を読まなければならない
- ○文書要求の条件を細分化し類似性を見つける
  - ▹「意味構造(係受け情報)の利用」，「共起情報の利用」

意味構造の利用(例:国立情報学研究所)

意味ネットワークで文を表現．係受け構造の一致をみる．

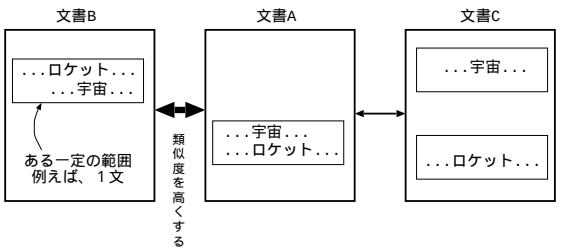


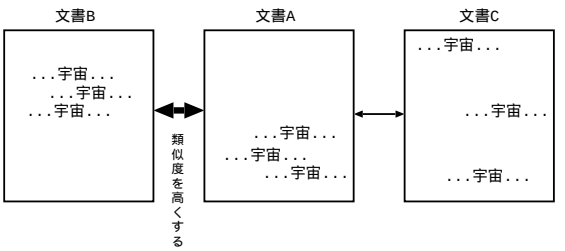
語の関係を利用する(1)

□語の共起

□語彙連鎖

語の共起(近距離に現れる語の組)



語の関係を利用する(2)

□語彙連鎖(同じや関連のある語の繰り返し)