【WWW2014勉強会】

Session 16: Content Analysis 2 - Topics

担当：白川（大阪大学）

2014年7月20日（日）

Session 16: Content Analysis 2 - Topics の論文

p527.pdf

A Time-Based Collective Factorization for Topic Discovery and Monitoring in News

Carmen Vaca (Politecnico di Milano & Escuela Superior Politecnica del Litoral), Amin Mantrach (Yahoo! Labs), Alejandro Jaimes (Yahoo! Labs), Marco Saerens (Université de Louvain)

P539.pdf

The Dual-Sparse Topic Model: Mining Focused Topics and Focused Terms in Short Text

Tianyi Lin (The Chinese University of Hong Kong), Wentao Tian (The Chinese University of Hong Kong), Qiaozhu Mei (University of Michigan), Hong Cheng (The Chinese University of Hong Kong)

P551.pdf

Acquisition of Open-Domain Classes via Intersective Semantics

Marius Paşca (Google Inc.)

Session 16: Content Analysis 2 - Topics の論文

p527.pdf

A Time-Based Collective Factorization for Topic Discovery and Monitoring in News

Carmen Vaca (Politecnico di Milano & Escuela Superior Politecnica del Litoral), Amin Mantrach (Yahoo! Labs), Alejandro Jaimes (Yahoo! Labs), Marco Saerens (Université de Louvain)

P539.pdf

The Dual-Sparse Topic Model: Mining Focused Topics and Focused Terms in Short Text

Tianyi Lin (The Chinese University of Hong Kong), Wentao Tian (The Chinese University of Hong Kong), Qiaozhu Mei (University of Michigan), Hong Cheng (The Chinese University of Hong Kong)

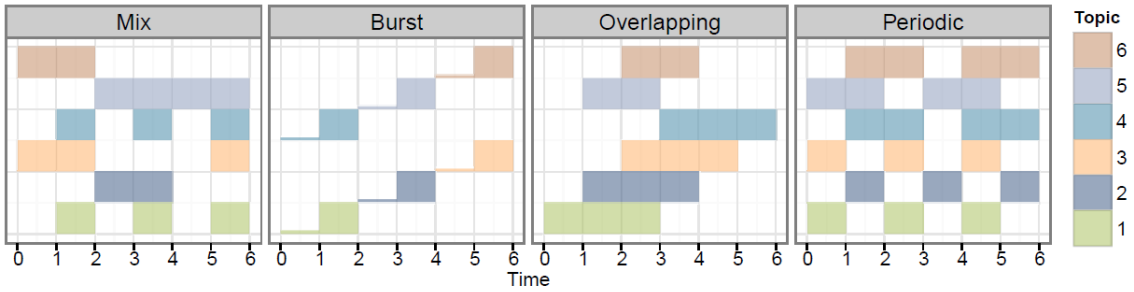
P551.pdf

Acquisition of Open-Domain Classes via Intersective Semantics

Marius Paşca (Google Inc.)

研究背景とか

- ニュースのトピックを各時刻ごとに検出し，モニタリングしたい
  - トピックの進化・出現・消失に対応しつつモニタリング



トピックの現れ方
いろいろ（論文より）

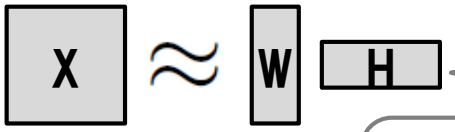
Figure 1: Synthetic dataset representing four scenarios.

- 今までにそういう研究なかった？
  - モニタリングなので，異なる時刻のトピックを「明示的に」リンクさせたい（i.e., 前のどのトピックが現在のどのトピックに繋がっているのか）
  - それに対応してるのが今のところTM-LDA [Wang, KDD12] ぐらいらしい
    - TM-LDA: トピック遷移行列を作成（i.e., このトピックが出たら次にこのトピックが出やすい）
    - これまでの傾向を学習するのみ，突発的な災害等のトピックはすぐには追えない
    - もっと現時点のデータからダイレクトにトピック遷移を発見したい！

提案手法

- Joint Past Present（JPP）分解
  - NMF (Non-negative Matrix Factorization) で文書-単語行列 X を
    文書-トピック行列 W とトピック-単語行列 H に分解（t は時刻）

$$\mathbf{X}^{(t)} \approx \mathbf{W}^{(t)}\mathbf{H}^{(t)}$$



イメージ図

  - さらにトピック遷移行列 M を用いてこれを表現

$$\mathbf{X}^{(t)} \approx \mathbf{W}^{(t)}\mathbf{M}^{(t)}\mathbf{H}^{(t-1)}$$

ひとつ前の時刻の H に
遷移行列 M をかけたら
今の時刻の H になる的な

  - この二つをそれぞれ制約とし，正則化項入れて解く
    - L は凸ではないが，ある程度いい局所解が得られる [Lee, Nature99]
    - 無理に $\mathbf{M}^{(t)}\mathbf{H}^{(t-1)}$ を $\mathbf{H}^{(t)}$に一致させたりはしない（過去に引っ張られてしまうため？）

$$L = \arg\min_{\mathbf{W}^{(t)},\mathbf{H}^{(t)},\mathbf{M}^{(t)}} \|\mathbf{X}^{(t)} - \mathbf{W}^{(t)}\mathbf{H}^{(t)}\|_F^2 + \|\mathbf{X}^{(t)} - \mathbf{W}^{(t)}\mathbf{M}^{(t)}\mathbf{H}^{(t-1)}\|_F^2 + \lambda\|\mathbf{M}^{(t)} - \mathbf{I}\|_F^2 + \alpha\|\mathbf{H}^{(t)}\|_1 + \beta\|\mathbf{W}^{(t)}\|_1 + \gamma\|\mathbf{M}^{(t)}\|_1$$

$\lambda \to \infty$ にすれば
過去指向
$\lambda \to 0$ にすれば
現在指向

アルゴリズム

- 割と短い，行列計算ライブラリ使えば簡単に実装できるかも

input : $\mathbf{X}^{(t)}, \mathbf{H}^{(t-1)}, \lambda, \epsilon$
output: $\mathbf{W}^{(t)}, \mathbf{H}^{(t)}, \mathbf{M}^{(t)}$

$\mathbf{W}^{(t)}, \mathbf{H}^{(t)}, \mathbf{M}^{(t)} \leftarrow$ *random non-negative init*;
$\delta^{'} \leftarrow maxInt, \delta \leftarrow \frac{\delta^{'}}{2}$;
$\beta \leftarrow$ *to choose in [0.001,0.05] [5]*;
$\lambda \leftarrow$ *to choose in* $[0, \infty[$;
while $abs(\delta^{'} - \delta) \geq \epsilon$ **do**

$$\mathbf{H}^{(t)} \leftarrow \mathbf{H}^{(t)} \odot \frac{[(\mathbf{W}^{(t)\mathrm{T}}\mathbf{X}^{(t)} - \alpha)]}{[(\mathbf{W}^{(t)\mathrm{T}}\mathbf{W}^{(t)}\mathbf{H}^{(t)})]};$$

$$\mathbf{W}^{(t)} \leftarrow \mathbf{W}^{(t)} \odot \frac{[(\mathbf{X}^{(t)}\mathbf{H}^{(t)\mathrm{T}} + \mathbf{X}^{(t)}\mathbf{H}^{(t-1)\mathrm{T}}\mathbf{M}^{(t)\mathrm{T}} - \beta)]}{[(\mathbf{W}^{(t)}(\mathbf{H}^{(t)}\mathbf{H}^{(t)\mathrm{T}} + \mathbf{M}^{(t)}\mathbf{H}^{(t-1)}\mathbf{H}^{(t-1)\mathrm{T}}\mathbf{M}^{(t)\mathrm{T}}))]};$$

$$\mathbf{M}^{(t)} \leftarrow \mathbf{M}^{(t)} \odot \frac{([\mathbf{H}^{(t-1)}\mathbf{X}^{(t)\mathrm{T}}\mathbf{W}^{(t)} + \lambda\mathbf{I} - \gamma)]}{[(\mathbf{H}^{(t-1)}\mathbf{H}^{(t-1)\mathrm{T}})\mathbf{M}^{(t)\mathrm{T}}(\mathbf{W}^{(t)\mathrm{T}}\mathbf{W}^{(t)}) + \lambda\mathbf{M}^{(t)\mathrm{T}}]};$$

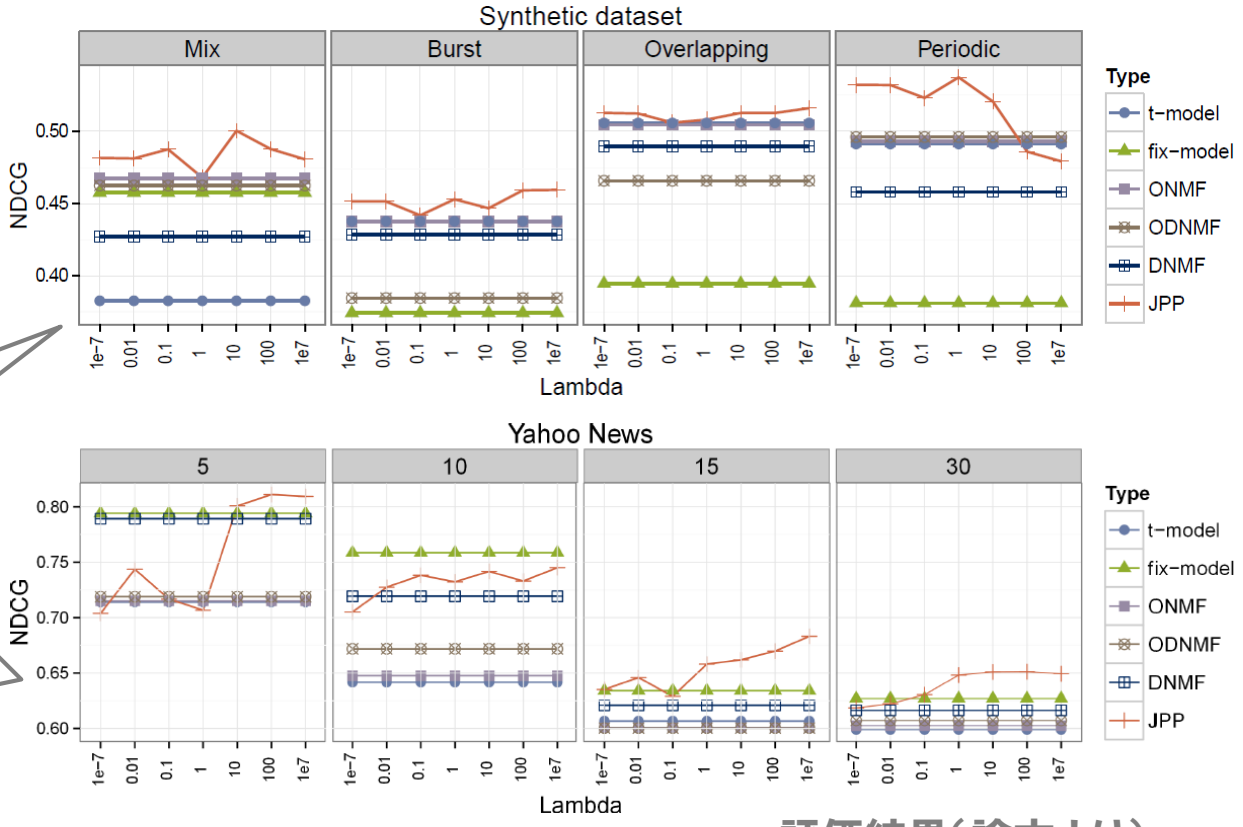
$\delta^{'} \leftarrow \delta$;
$\delta \leftarrow \mathcal{L}(\mathbf{X}^{(t)}; \mathbf{W}^{(t)}; \mathbf{H}^{(t)}; \mathbf{M}^{(t)}; \mathbf{H}^{(t-1)})$;
end

Algorithm 1: Joint Past Present decomposition Algorithm.

JPP分解アルゴリズム（論文より）

評価 – トピック検出

- **リアルデータと人工データの両方で評価**
  - リアル：Yahoo News，13,319記事，76種類のカテゴリタグ
  - 人工：TDT2データをMix, Burst, Overlapping, Periodic用に改造，6トピック
- **比較手法**
  - NMFのいろんなバリエーション
    （TM-LDAとは比較しないのか…）



人工データでは結構強い

λ が大きいとPeriodicで性能低下

リアルデータではトピック数が多い（15以上）場合に有効

λ が大きい（過去の情報を参考にする）ほうが良さげ

評価結果（論文より）

評価 – トピックトラッキング

- **時間的にトピックのつながりを追っかけたいって話でしたね**
  - 過去のトピック-単語行列 H を用いて現在の文書-トピック行列 W を予測
    → 既知のトピックが未来の各時刻でどの文書に出てくるかを見る
  - Online-NMF（ONML）[Cao, IJCAI07]とのみ比較 （やはり比較されないTM-LDA…）

Online NMFにはなんとか勝ってるがこれでいいのか！？

k	Alg.	microF1	macroF1	MAP	NDCG
15	NMF	0.56 ± 0.02	0.54 ± 0.02	0.68 ± 0.02	0.80 ± 0.01
	JPP	**0.58** ± 0.02	**0.56** ± 0.02	**0.70** ± 0.02	**0.81** ± 0.01
30	NMF	0.45 ± 0.04	0.42 ± 0.03	0.58 ± 0.03	0.73 ± 0.02
	JPP	**0.48** ± 0.03	**0.46** ± 0.03	**0.61** ± 0.03	**0.75** ± 0.02

Table 4: Topic classification evaluation on the Yahoo News data set using four metrics. The bold-faced numbers indicate that JPP is significantly better than the other methods (p value <0.01 in Wilcoxon paired test). The values are averaged over 21 folds.

評価結果（論文より）

Session 16: Content Analysis 2 - Topics の論文

p527.pdf

A Time-Based Collective Factorization for Topic Discovery and Monitoring in News

Carmen Vaca (Politecnico di Milano & Escuela Superior Politecnica del Litoral), Amin Mantrach (Yahoo! Labs), Alejandro Jaimes (Yahoo! Labs), Marco Saerens (Université de Louvain)

P539.pdf

The Dual-Sparse Topic Model: Mining Focused Topics and Focused Terms in Short Text

Tianyi Lin (The Chinese University of Hong Kong), Wentao Tian (The Chinese University of Hong Kong), Qiaozhu Mei (University of Michigan), Hong Cheng (The Chinese University of Hong Kong)

P551.pdf

Acquisition of Open-Domain Classes via Intersective Semantics

Marius Paşca (Google Inc.)

研究背景とか

- ツイートなどの短文にも強いトピックモデルを作りたい
  - 短文では語の共起がスパースなので通常のLDAだと性能落ちる

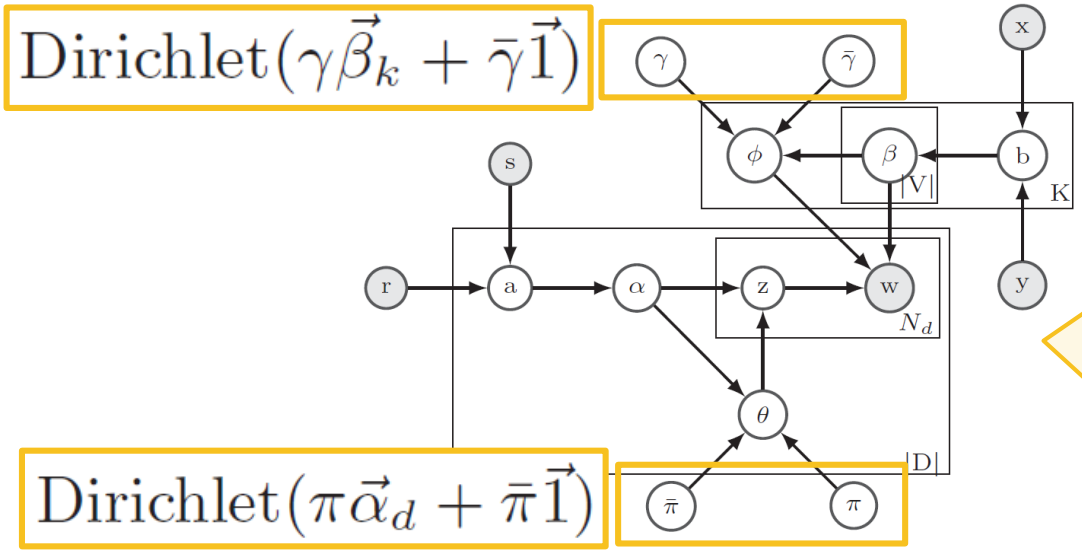
- 今まで改善策あったのでは？
  - ヒューリスティックな方法でLDAを改善する研究がある
    - Document pooling [Mehrotra, SIGIR13], Contextualization [Tang, KDD13]
    - ツイート投稿者など，テキスト以外のAdditionalな情報が必要
  - Sparsity-Enhanced Topic Modelと呼ばれる研究もある
    - “sparsity-enhanced topic models ... [27, 8, 29, 23, 30, 1, 32, 14, 28, 17].”
    - めっちゃあるやないか！
    - 有力候補1：STC（Sparse Topical Coding）[Zhu, UAI11]
      欠点：文書のスパースなトピック表現（1文書は少数のトピックのみを持つ）ができてない
    - 有力候補2：IBP-LDA [Archambeau, NIPS Workshop 11]（IBP: Indian Buffet Process）
      欠点：複雑なので大規模な文書集合には適用できない

多いので論文中の引用番号のみ

実際の短文はこの性質があるらしく，上記を実現したほうが性能が良くなる

提案手法

- Spike and Slab prior [Ishwaran, The Annals of Statistics 05] を入れる
  - 「1文書は少数のトピックのみを持ち，1トピックは少数の単語のみを持つ」という（特に短文に当てはまる）性質を表すのに適したprior（事前分布）
  - 弱いスムージングprior（＝より頻度主義に近くなる）でこれを表現しようとしても，短文の疎な部分がそのまま出てくるだけ [Wang, NIPS09]
  - なのでSpikeとSlabのトピックそれぞれに別々のスムージングをかける
    → Spikeで一部のトピックを際立たせつつ，Slabで滑らかに！



SpikeとSlabの和で表されたハイパーパラメータでトピックと単語それぞれのディリクレ分布を生成

Figure 1: The graphical model of DsparseTM

グラフィカルモデル（論文より）

LDAの更新式

- Zero-Order Collapsed Variational Bayes Inference（CVB0）[Asuncion, UAI09]
  - Zero-Order：更新式のテイラー展開の0番目のオーダのみで近似
  - 理論的に安定していることが証明されている [Sato, ICML12]
  - Collapsed Gibbs Sampling [Griffiths, PNAS04]やCollapsed Variational Bayes Inference（CVB） [Teh, NIPS06]より性能良い
  - 処理時間も短縮できるので，大規模なデータに向いている

LDAの更新式は今のところCVB0使っとけば良さげ（私の個人的な感想です）

評価

- DBLP, 20 Newsgroups, Twitterの3種類のデータ

Table 2: Statistics of the data sets

Data set	# Documents	Vocabulary size	Avg doc len by words
DBLP	40,190	9,393	5.7
20NG	18,774	60,698	114.8
TWITTER	1,119,464	32,641	4.9
TWITTER-A	13,080	15,952	451.4

データセット(論文より)

ユーザごとにツイートをまとめたもの

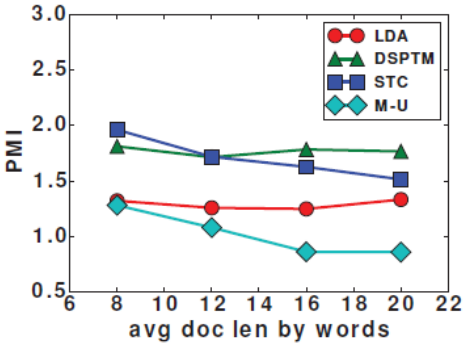
提案手法(DsparseTM)が全般的に良い

Twitterだとほぼ1ツイート1トピックらしく，Mixture of unigrams [Blei, JMLR03] がベストだが，それでも提案手法は結構良い

20 Newsgroupsの文書長を短くした場合でも提案手法(DSPTM)が安定

Table 3: Topic coherence (PMI) on four data sets

	DBLP	20NG	TWITTER	TWITTER-A
Number of topics	15	120	200	200
DsparseTM	**0.871**	**1.621**	1.051	**1.939**
LDA	0.622	1.336	0.562	1.757
STC	0.088	1.515	0.378	1.192
Mixture of unigrams	0.532	0.691	**1.121**	0.823
Conference topic	0.586	-	-	-



評価結果(論文より)

Session 16: Content Analysis 2 - Topics の論文

p527.pdf

A Time-Based Collective Factorization for Topic Discovery and Monitoring in News

Carmen Vaca (Politecnico di Milano & Escuela Superior Politecnica del Litoral), Amin Mantrach (Yahoo! Labs), Alejandro Jaimes (Yahoo! Labs), Marco Saerens (Université de Louvain)

P539.pdf

The Dual-Sparse Topic Model: Mining Focused Topics and Focused Terms in Short Text

Tianyi Lin (The Chinese University of Hong Kong), Wentao Tian (The Chinese University of Hong Kong), Qiaozhu Mei (University of Michigan), Hong Cheng (The Chinese University of Hong Kong)

P551.pdf

Acquisition of Open-Domain Classes via Intersective Semantics

Marius Paşca (Google Inc.)

研究背景とか

※ 2014年7月11日時点

- ドメイン非依存で細粒度なクラスを獲得したい
  - クラス獲得タスクでは，クラスのラベル名を獲得するだけでなく，クラス（例：tech company）とインスタンス（例：Google）の紐づけも必要
  - よくある手法：「such as」の前後に出現する語をWebのテキストから抽出

Hearst patternと呼ばれる

- 問題：詳細なクラスだとテキストに直接現れることがレア
  - 例：「gold mining companies listed on the toronto stock exchange」
  - 上のクエリでのGoogle検索結果※：3件（うち1件はこの論文）
  - クラスと紐づけされるインスタンスが同時に出現することはさらに激レア

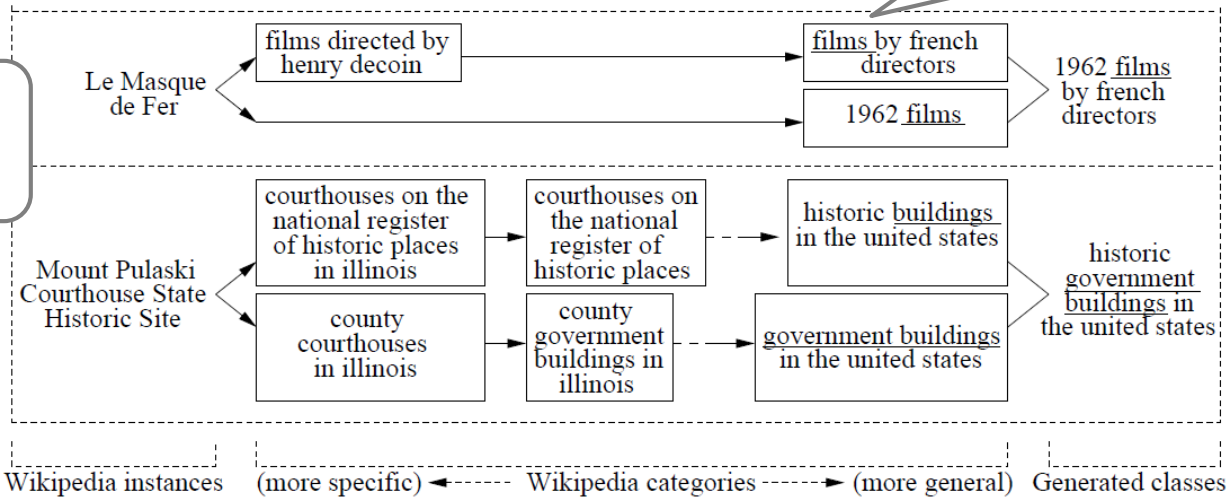
つまり，細粒度なクラスに関する情報はデータ量に頼った方法（＝Webテキスト全走査）でも得られない
天下のGoogleさんが言うんだから間違いない

提案手法

- 二つのクラスを組み合わせてより詳細なクラスを生成する
  - 「gold mining companies」
    ＋「companies listed on the toronto stock exchange」
    ＝「gold mining companies listed on the toronto stock exchange」
  - あるインスタンスが二つのクラスに属する場合，このインスタンスは組み合わさったクラスにも属する
  - 組み合わさったクラスのラベルがクエリログに含まれていれば採用

共通するヘッドワードを軸に組み合わせる

情報検索への応用を意識



Wikipediaに適用した場合（論文より）

評価 – 規模

- 2012年5月の英語版Wikipediaで評価
  - 2億のWebページ(N-gramモデル)と10億の検索クエリを補助的に利用
  - R_W: Wikipediaカテゴリ(クラス)のラベル
    $R_{G \cap \neg W}$: 提案手法で得られたクラスのラベル(Wikipediaカテゴリ除く)
    R_Q: 検索クエリに含まれるラベル
    R_S: 拡張された検索クエリ(1250億)に含まれるラベル [Paşca, CIKM11]
    N-gramモデル使って疑似的に
    似たようなクエリめっちゃ生成する

Run	Class Labels	I per C		Run	Class Labels	I per C	
		A	M			A	M
R_W	472,237	310	8	$R_{G \cap \neg W}$	4,576,369	10	2
$R_{W \cap Q}$	136,259	579	15	$R_{G \cap \neg W \cap Q}$	33,207	35	3
$R_{W \cap S}$	272,694	416	10	$R_{G \cap \neg W \cap S}$	389,878	27	2

Table 1: Number of (unique) class labels extracted in various runs (I per C=number of instances per class label; A=average; M=median)

ラベル数めっちゃ増える

クエリの制約を入れるとだいぶ減るが，それでも結構増える

得られたクラスのラベル数(論文より)

評価 – 精度

- **ランダムにサンプルとって人手で評価**(この分野，今のところこれがベストな評価方法)
  - クエリの制約入れると9割ぐらいの精度出る

Run	Precision				
	Correctness Counts				Score
	T	C	Q	I	
$R_{G \cap \neg W}$	200	119	45	36	0.707
$R_{G \cap \neg W \cap Q}$	200	172	6	22	0.875
$R_{G \cap \neg W \cap S}$	200	179	1	20	0.897

Table 5: Accuracy computed over random samples of pairs of a generated class label that is not a Wikipedia category, and one of the instances of the class label. For each run, the sample of pairs is drawn by first selecting a weighted random sample of 200 class labels, where the weight is the size of (i.e., number of instances in) a class label; then selecting one random instance for each class label (T=total; C=correct; Q=questionable; I=incorrect)

精度評価(論文より)

評価 – その他

- 他にもいろいろ評価している(評価メインの論文)
  - すでにWikipediaにあるインスタンス(正解)がちゃんと取れてるか?
  - Wikipediaで定義されていないクラス-インスタンス間の関係を新たにどれだけ取れるか?
  - 規模や精度についての様々な観点での評価
  - エラーについての考察
  - などなど

この先はぜひ自分の目で確かめてほしい!