検索実験の方法と実際：NTCIR ワークショップでの試み

Methodology and Pragmatics of Retrieval Experiments at NTCIR Workshop

岸田和明 †　岩山真 ‡　江口浩二*
Kazuaki KISHIDA†　Makoto IWAYAMA‡　Koji EGUCHI*
† 駿河台大学／国立情報学研究所 kishida@surugadai.ac.jp
‡ 東京工業大学／日立製作所 iwayama@pi.titech.ac.jp
＊国立情報学研究所 eguchi@nii.ac.jp

2002 年 10 月 8 日
Pre-meeting Lecture at the NTCIR-3 Workshop（於：学術総合センター）

概 要

The paper aims at describing purpose, methods or research problems of information retrieval experiments at the NTCIR workshop, a research project on information retrieval in Japan. First, we focus on some general issues on test collection, relevance concept, indicators for evaluation, retrieval models, and implementation of retrieval system. Second, each purpose and methodology of three retrieval tasks at the NTCIR-3 workshop, cross-language retrieval task, patent retrieval task and web retrieval task, is described respectively.

1　はじめに

情報検索（information retrieval：IR）の研究・開発は 1950 年代から 60 年代にかけて本格化し，オンライン検索システムや CD-ROM 検索システム，さらには最近のインターネットのサーチエンジン等の開発・普及を経て，今日に至っている．この発展に**検索実験**（retrieval experiment）の果たした役割は大きい．検索実験に基づく実証分析なしには，これまでの情報検索技術の発展はなかったと言っても過言ではない．

最も初期の頃の検索実験としては，英国における Cranfield 実験が有名である．これは，1950 年代から 60 年代にかけて，主として自由語と統制語の性能を比較するために実施されたもので，この研究を通じて，現在の検索実験の方法論的な基盤が形成された．

その後，様々な検索実験が試みられてきたが，その中でも特筆すべきは，1990 年代の前半に開始された，米国における大規模検索実験プロジェクト TREC（Text REtrieval Conference）[1] である．このプロジェクトは，それまでの検索実験とはその規模や内容の点で一線を画しており，近年の情報検索技術の発展に多大な貢献を果たすことに成功した．一般的には，インターネットの普及・発展によって情報検索への関心がさらに高まっており，TREC が今後も重要な役割を果たすことが期待される．

TREC の成功に刺激を受けて，同様な検索実験プロジェクトが日本でも開始された．１つは 1998 年から 1999 年にかけて実施された IREX[2] であり，もう 1 つは，国立情報学研究所（当時は学術情報センター）によってやはり 1998 年に開始され，現在続行中の NTCIR（NII/NACSIS Test Collection for

[1] http://trec.nist.gov/
[2] http://cs.nyu.edu/cs/projects/proteus/irex/

Information Retrieval）プロジェクト[3] である [1].

TREC と比較した場合の NTCIR の大きな特徴は，主として検索対象を日本語で書かれたテキスト（あるいは文書）としている点である．日本語に対する情報検索には未解決な問題が多く，さらなる技術発展が必要とされている．このためには，情報検索分野の研究者のほかにも，自然言語処理やデータベース管理システムの研究者，さらには図書館情報学の研究者などが NTCIR に参加して，テキスト処理に関する総合的な研究・開発に取り組むことが重要である．なお，最近の NTCIR では，中国語や韓国語といった東アジアの諸言語も対象となっており，これらの言語に対する研究もまた必要であることは言うまでもない．

本資料は，これから NTCIR に参加しようと計画している人たちに向けて，検索実験の基礎的なことがらや，初歩的な検索システム構築のための知識を解説することを目的としている．この資料では，以下，第 2 節で検索実験の概要を説明し，続いて第 3 節で主な検索モデルについて述べる．さらに実験的な検索システムの構築方法を第 4 節で解説し，第 5 〜 7 節では，2001 年に始まった NTCIR-3 のプロジェクトにおける 3 つの検索タスク（言語横断検索，特許検索，Web 検索）についてそれぞれ説明する．最後に，第 8 節で利用可能な研究資源に関して簡単に触れる．

2 検索実験

2.1 NTCIR 小史

NTCIR は TREC における方法論を模倣するかたちで，1998 年に開始された．TREC と同様に，成果報告会（ワークショップ）をひとつの区切りとして，何回かの実験を繰り返しており，現在までに，

- NTCIR-1：1998 年 11 月〜 1999 年 9 月
- NTCIR-2：2000 年 6 月〜 2001 年 3 月
- NTCIR-3：2001 年 8 月〜 2002 年 10 月

の合計 3 回の実験が実施されている．

このため，上に示したように，各回を区別する目的で,「NTCIR」という名称の後ろに「1」「2」などの回数を示す数字を付けて表記することが多い．各回の実験では，いくつかの個別的な研究課題が設定され，**タスク**（task）と呼ばれている．各回で設定されたタスクを表 1 に示す（詳しくは文献 [2] 参照）．

表 1: NTCIR ワークショップのこれまで

回	タスク	チーム数
1	随時検索	18
	言語横断検索	10
	用語抽出	9
2	中国語検索（言語横断含む）	11
	日本語・英語検索（同上）	25
	テキスト要約	9
3	言語横断検索	31
	特許検索	15
	質問応答	18
	テキスト要約	11
	Web 検索	13

2.2 テストコレクション

検索実験をおこなうには，**テストコレクション**（test collection）が必要である．テストコレクションは，一般に，

- 検索対象となる文書集合（データベース）
- 検索質問の集合
- 各検索質問に対して各文書が適合しているかどうかの情報

の 3 つの要素から構成される．**文書**（document）とは，文字から構成される**テキスト**（text）を伴った，論理的あるいは物理的な単位である．このような文書の集合をデータベースとして組織し，利用者からの**検索質問**（search request）に対して，効率的かつ効果的に文書を検索できるようにすることが，情報検索の基本的問題である．したがって，この場合の検索は，より正確には，**文書検索**（document retrieval）であり，また，**テキスト検索**（text retrieval）と呼ばれることも多い．

検索実験に使われる検索質問は，特に**検索課題**と呼ばれることがあり，実際の利用者から収集されたり，あるいは人工的に作成される．検索実験では，これらの検索課題に**適合する**（relevant）文書を文書集合中であらかじめ特定しておき，それらを実際

[3] http://research.nii.ac.jp/~ntcir/

にうまく検索できるかどうかという観点から，そのシステムを評価することになる．なお，検索実験の場合，このような前もって特定された適合文書を**正解文書**と呼ぶことがある．

NTCIR-1で使用された文書の例を以下に示す．この場合の文書は，学会発表に関する，抄録付きの書誌情報である．

<REC>
<ACCN>gakkai-j-0000441590</ACCN>
<TITL>大規模テストコレクション NTCIR-1の構築（1）ープーリングと正解判定の分析ー</TITL>
<AUPK>栗山 和子 / 江口 浩二 / 野末 俊比古 / 神門 典子</AUPK>
<CONF>全国大会</CONF>
<CNFD>1999. 09. 28 - 1999. 09. 30</CNFD>
<ABST>本研究の目的は,（1）大規模テストコレクションを構築する手法としてのプーリングの有効性を検証し,（2）プーリング件数が検索システムの評価に関連があるかどうか調べ,（3）正解判定の際の判定のゆれがシステムの評価に関係してくるかどうかを明らかにすることである．...（中略）... 検索結果を評価したとき，検索精度の平均は異なる正解判定リスト間においてほとんど差がなくなり，多数の検索課題を用いて評価を行なえば，判定者間の判定のゆれは評価においては問題ではないということがわかった．</ABST>
<SOCN>情報処理学会</SOCN>
</REC>

次に，NTCIR-3 の CLIR タスクにおける検索課題の例を示す．この検索課題には，NUM，SLANG，TLANG，TITLE，DESC，NARR，CONC の 7 つのフィールドが含まれているが，このうち標準的なフィールドは最後の 4 つであり，それぞれ

- TITLE（title）：検索課題の内容を簡単に表すタイトル
- DESC（description）：検索する内容を文で記述したもの
- NARR（narrative）：検索する内容の詳細な説明
- CONC（concept）：検索する内容を表すキーワード

である．なおタグ名称はタスクによって異なることがある．

<TOPIC>
<NUM>013</NUM>
<SLANG>CH</SLANG>
<TLANG>EN</TLANG>
<TITLE>NBA labor dispute</TITLE>
<DESC>To retrieve the labor dispute between the two parties of the US National Basketball Association at the end of 1998 and the agreement that they reached.</DESC>
<NARR>The content of the related documents should include the causes of NBA labor dispute, the relations between the players and the management, main controversial issues of both sides, compromises after negotiation and content of the new agreement, etc. The document will be regarded as irrelevant if it only touched upon the influences of closing the court on each game of the season.</NARR>
<CONC>NBA (National Basketball Association), union, team, league, labor dispute, league and union, negotiation, to sign an agreement, salary, lockout, Stern, Bird Regulation.</CONC>
</TOPIC>

現実の検索の状況では，伝統的なオンライン検索にせよ，インターネットのサーチエンジンにせよ，通常，少数の検索語がシステムに投入される傾向にある．それに対して，検索実験では，多くの場合，description や narrative のような，かなり長いテキストも用意される．テストコレクションを利用する研究者は自分の研究目的に応じて，上記のフィールドを取捨選択できる．例えば，現実により近い検索を想定するならば title のみを使えばよいし，検索質問を文章で入力させるようなシステムの開発を想定するならば，description を使用することになる．なお，NTCIR の検索実験では，比較のため，検索実

験に参加するチームに対して，必ず description のみを使用した実行結果を提出するよう求めている．

2.3 評価尺度と適合判定

2.3.1 適合性の概念

上で述べたように，検索実験では,「文書集合中に含まれる適合文書（正解文書）をどれだけうまく特定できるか」という観点からの評価がなされる．もちろん，実際に検索システムを開発する場合，応答速度や必要な資源の量などの他の観点も重要ではある（むしろこれらのほうが重視される場合も少なくない）．しかし，TREC や NTCIR のような検索実験では，それらは副次的に考慮されるのみであり，適合文書の検索という点からの有効性（effectiveness）が評価のための第一の尺度となっている．

適合性（relevance）とは，検索実験においては,「検索課題における主題と文書における主題とが一致していること」を意味し，その程度は**適合度**と呼ばれる．この適合性の概念については，図書館情報学を中心に，その定義や妥当性をめぐって数多くの哲学的な議論がなされている [3]．この議論に照らせば，上の適合性の定義はそれほど厳密ではないが，実際的には十分である（通常，実際の適合判定では，もう少し具体的な操作的定義が作業マニュアルの中で与えられる）．ここではこの議論には深く立ち入らず，適合性に類似した概念をいくつか列挙するのに留めておく．

- 適切性（pertinence）：検索質問ではなく，それが作成される元となった，利用者の情報要求（information needs）に対しての文書の内容の一致性
- 情報性（informativeness）：検索によってもたらされた情報が利用者にとってこれまでに知らなかったものであるという意味での新奇性（novelty）を持ち，なおかつ，その情報を利用者が理解可能であること
- 有用性（usefulness）：もたらされた情報が，利用者に対して実際的な価値を持つこと

適切性を含めて，これらはある特定の実際の利用者が存在してはじめて意味を持つ性質であり，科学的な実験環境において，このような性質を正確に測定し，なおかつ客観的に評価することはきわめて難しい．そのため，検索実験では，評価のための尺度として，実際の利用者の介在なしに測定可能な「適合性」の概念が一般的に用いられている．ただし，これは，検索実験が適切性・情報性・有用性などの観点からの評価自体を否定していることを意味するわけではない．

2.3.2 プーリングの方法

文書集合の中から適合文書（正解文書）を洗い出す作業は容易ではない．例えば，10 万件の文書から成るデータベースの場合，これらをすべて調べて，正解文書を見つけ出すことは困難である．実際には，10 万件の文書をすべて調べるのは不可能であり，TREC 以前のテストコレクションにおける文書集合が十分な大きさを持たなかった原因はこの点にある．

TREC では，**プーリング**（pooling）と呼ばれる方法を使ってこの問題を解決し，現実のデータベースの規模に匹敵する実験用文書集合を提供することに成功した．その基本的なアイデアは，検索実験に数多くの研究チームを参加させ，1 つの検索課題に対するそれぞれの検索結果を併合して（プールして），その文書集合に対してのみ，正解文書を調べるというものである．この方法ならば，調べる文書は格段に少なくて済む．もちろん，発見されない正解文書が「もれ」として存在する可能性はあるが，参加した研究チームのシステムがそれなりの性能を発揮していれば，近似的な評価方法としては十分である[4]．

近年の検索実験における研究の主対象は，**適合度順出力**である．これは，システムが検索質問に対する各文書の適合度を推測し，その適合度に従って順位を付けて文書を出力する検索様式である（すなわち ranked output）．この場合には，各チームの出力結果の上位の何件かだけをプールして，それらのみを調べればよい．その結果，作業としては，例えば，

1. 1 つの検索課題に対する実行結果として上位 1,000 件を各チームに提出してもらい，
2. そのうち上位 100 件ずつを抽出して，プールに

[4]逆に言えば，参加者が少なく，かつそれらのシステムの性能に期待できない場合には，問題が生じることになる

入れる

ということになる．例えば 20 チームが参加して，それぞれ実行結果を 1 つずつ提出したとすれば，プールに含まれる文書は 2,000 件よりも少なくなる（重複分が除かれるためである．なお，実際には，1 つのチームが複数の実行結果を提出している）．この程度の規模の文書集合ならば，正解文書の洗い出しはそれほど難しい作業ではない．抽出する文書を少なくするか（いわゆる「浅い」プーリング），多くするか（「深い」プーリング）にするかは，検索課題や文書集合，参加チームの状況などに依存する [4, 5]．

2.3.3 適合判定とその問題

正解文書（適合文書）を確認する作業は，**適合判定**（relevance judgment）と呼ばれる．実際に検索課題を作成した人が適合判定をおこなう場合もあれば，その他の第三者が判定することもある．

適合判定では，各検索課題に対して，2 値または多段階の適合度を各文書に付与する．2 値の場合には，適合（relevant）／不適合（irrlevant）であり，多段階の場合には，例えば，高適合（highly relevant），適合（relevant），部分適合（partially relavant），不適合（irrelevant）などのように設定する．現在では，評価指標（後述）の多くは 2 値での適合判定に対応したものであり，そのため，多段階での判定結果は 2 値に圧縮される．例えば，高適合と適合を「適合」，その他を「不適合」としたり，あるいは，部分適合以上を「適合」とする方法などがある．2 値に圧縮せずに，多段階の判定結果をそのまま利用する評価指標の研究も進められており [6]，今後，一般的に利用される可能性がある．

適合判定の作業にはかなりの労力が必要であり，その客観性を維持することは難しい．このため，判定者の読み違いやかん違いなどに伴う誤りが生じる可能性がある．NTCIR プロジェクトでは，適合判定用のシステムに工夫を加え，この種の誤りを最小限に抑える努力が払われているが，それでも過誤が生じる可能性を完全には否定できない．そもそも適合性の判断は高度に主観的な作業であり，本質的に確率的な事象である（すなわち，一種の測定誤差が混入せざるを得ない）という解釈も成り立ちうる．

したがって，上で述べたプーリングに起因する問題も含めると，適合判定による評価には，

- 発見されない適合文書に伴う誤差
- 適合判定における測定誤差

の 2 種類の誤差が含まれうることになる．しかし，この誤差による評価結果の偏り（bias）は，複数の検索課題の各々に対して計算される評価指標の値の平均をとることによって，相殺されることが期待される．なぜなら，上記の誤差がある特定のシステムだけに対して常に有利または不利に働くことは考えにくく，その結果，平均的には偏った影響を与えないと想定されるからである（逆に言えば，ごく少数の検索課題のみを取り上げて議論する場合には注意しなければならない）．

2.4 性能評価のための指標

2.4.1 精度と再現率

検索が成功したかどうかを評価するための伝統的な指標は，**精度**（precision）と**再現率**（recall）である．なお，精度に対しては，「適合率」という用語が使用される場合がある[5]．

検索課題を 1 つに固定する．文書集合中に含まれる，この課題に対する適合文書の総数を R とかく．また，あるシステムを使って検索を実行し，ある基準に従って n 件の文書を出力したとき，それに含まれる適合文書の数が r 件であったとする．このとき，精度は r/n，再現率は r/R で定義される．

一般に精度と再現率はトレードオフの関係にある．すなわち，精度が上がるようにシステムのパラメータを調整すると，再現率は下がってしまい，逆に再現率を上げようとすると精度が下がる．数学的には，調整後に新たに出力される適合文書の増加率が，同時に出力される不適合文書の増加率を下回るとき，トレードオフの関係が成立する [3]．

指標が 2 種類あると比較しにくいので，単一の数値を使いたいことがある．この場合には，基本的には精度と再現率の平均を計算すればよい．情報検索の分野では，このために調和平均を使うことが多い．これを F **尺度**と呼ぶ．精度を X，再現率を Y とす

[5] これは本来英語の relevance ratio に対応しているが，relevance ratio 自体は現在ではほとんど使われていないようである．

れば，$F = 2/(X^{-1} + Y^{-1})$ である．調和平均には，X がある程度高い状況では，X の増加よりもむしろ Y の増加のほうが平均値の増加に寄与するという性質がある（X と Y を入れ替えてもこの議論は当然成り立つ）．これは，F 尺度には，精度（再現率）がある程度満足された状況では，精度（再現率）の上昇よりもむしろ再現率（精度）の上昇のほうが利用者の満足に寄与する」という仮定が置かれていることを意味する．なお，この指標は，精度と再現率の重み付き調和平均を利用した E 尺度（文献 [3] 参照）を修正したものとして捉えることができる．

2.4.2 平均精度

精度や再現率は，適合度による順位付けをおこなわない伝統的なブール型検索が主流の時代において提案されたものである．したがって，適合度順出力を評価するには，いくつかの工夫を加える必要がある．適合度順出力の評価のために，TREC を中心とした検索実験で使用されている主要な指標は次のとおりである [7]．

- 精度（λ）：上位 λ 件の文書集合における精度
- 再現率（λ）：上位 λ 件の文書集合における再現率
- 再現率 50 %での精度：最上位の文書から順に調べていき，全適合文書の半分が出現した時点で計算した精度
- R 精度：上位 R 件の文書集合における精度（R は適合文書総数）
- 平均精度：最上位の文書から順に調べ，適合文書が出現した時点でそれぞれ精度を計算し，最後にそれらを平均したもの（実際には R で割る）

これらはいずれも非補間（non-interporated）の指標である（補間に関しては後述）．

TREC 等で標準的に使用されているツールである trec_eval [6] を使うと，精度（λ）や R 精度，平均精度などを計算できる．ただし，$\lambda = 5, 10, 15, 20, 30, 100, 200, 500, 1000$ である．

この中では，**平均精度**（average precision）が利用されることが多い．平均精度は，厳密には，次のように定義される．まず，順位 i 位の文書が適合しているならば 1，そうでなければ 0 となる変数を z_i とする．このとき，平均精度 v は，

$$v = \frac{1}{R} \sum_{i=1}^{n} \frac{z_i}{i} \left(1 + \sum_{k=1}^{i-1} z_k \right) \tag{1}$$

である [8]．R と n はこれまでどおり，それぞれ適合文書の総数，出力文書数である．平均精度の性質は詳しく調べられており，上で述べたような誤差に対して安定した評価指標であることが知られている [9, 8]．

例えば，R 精度と平均精度とを比較した場合，R の大きさにもよるが，通常，R 精度のほうが，文書の順位付きリストの上位の部分に限定して評価することになる（上位 R 件しか計算対象にならないため）．それに対して，平均精度では，より下位の部分に適合文書が存在すれば，それも計算対象に含められる．この意味で，平均精度のほうがマクロ的な指標である．ただし，平均精度の場合でも，上位での順位の変動のほうが下位におけるそれよりも値の変化に大きな影響を与える．例えば，1 位の適合文書と 2 位の不適合文書の順位が入れ替わった場合の平均精度の減少幅は，999 位の適合文書と 1000 位の不適合文書が入れ替わった場合のそれよりも大きい．

平均精度を使って，2 つの手法 A と B の性能に差があるかどうかを調べたいとする．このためには，MAP（mean average proecison）を計算しなければならない．MAP は各検索課題ごとの平均精度の平均を意味する．具体的には，検索課題が全部で L 件あり，それぞれの課題に対するあるシステムの平均精度を v_h と表記すれば $(h = 1, ..., L)$，その平均，

$$\bar{v} = \frac{1}{L} \sum_{h=1}^{L} v_h \tag{2}$$

が MAP に相当する．

経験的に，両者の MAP に 0.05 程度の差があれば，有意差があると言われているようであるが，より厳密には，平均値の差の検定を実行しなければならない[7]．ただし，同一のテストコレクションを使っ

[6] ftp://ftp.cs.cornell.edu/pub/smart/trec_eval.v3beta.shar

[7] この検定においてどのような母集団を想定するかについては注意を要する．現実に存在するあらゆる検索質問を母集団として考えることはおそらく無理である．テストコレクションとして用意されている検索課題の特徴に注意して，どのような検索質問集合の「代表」として捉えることができるかを慎重に検討してみる必要がある．

て手法 A と B を比較する場合，同一の検索課題に対して，手法 A と B のそれぞれの平均精度が計算されるため，平均値の差の検定は,「対になっているデータ（paired data）に対する検定」になる[8]．まず，h 番目の課題に対する手法 A，B の平均精度をそれぞれ v_{Ah}，v_{Bh} と表記する．2 つの手法の MAP の差 $\bar{v}_A - \bar{v}_B$ は，

$$\begin{aligned} \bar{v}_A - \bar{v}_B &= \frac{1}{L}\sum_{h=1}^{L} v_{Ah} - \frac{1}{L}\sum_{h=1}^{L} v_{Bh} \\ &= \frac{1}{L}\sum_{h=1}^{L} (v_{Ah} - v_{Bh}) \\ &= \frac{1}{L}\sum_{h=1}^{L} u_h \end{aligned} \tag{3}$$

である．ここで，$u_h \equiv v_{Ah} - v_{Bh}$ とおいた．したがって，この場合の平均値の差の検定は，変数 u_h の母平均が 0 であるという帰無仮説に対する検定問題に帰着する．このため，MAP にどれだけの差があれば，手法 A と B との性能に統計的有意差があるといえるかという問題は，変数 u_h の母分散の大きさと検索課題数 L に依存することになる．当然，この値は未知であるが，正規母集団を仮定して，u_h の標本分散 s_u を使って t 検定を実施すればよい（表計算ソフトや統計パッケージを使用すれば簡単である）．

具体的には，

$$y = \sqrt{s_u^2/L} \times P_t^{-1}(\alpha, L-1) \tag{4}$$

で計算される y よりも両手法の MAP の差が大きければ，有意水準 α で統計的有意差があると結論できる．ここで，$P_t^{-1}(\alpha, L-1)$ は自由度 $L-1$ の t 分布の逆関数[9]である（測定誤差を考慮したより詳細な議論については [8] 参照）．

2.4.3 再現率－精度グラフ

平均精度（あるいは MAP）は単一の数値なので比較等に便利であるが，もう少し詳細に分析したい場合には，精度（λ）を使えばよい．しかし，検索課題によって適合文書数 R が異なるため，精度（λ）は検索課題間での比較評価には不都合な場合がある．このようなときに，再現率を 0.1 で刻んだ 11 個の点（0.0,0.1,...,1.0）ごとの精度を算出できれば便利である．この方法ならば，例えば，$R = 200$ と $R = 15$ の検索課題間での比較が可能になる．

問題は，R の大きさによっては，11 個の点での精度が正確に決まらないことである．例えば，$R = 10$ ならば，$1/10 = 0.1$,$2/10 = 0.2$,... のように各点の値が正確に決まるが，$R = 11$ の場合，$1/11 = 0.0909...$，$2/11 = 0.1818...$ となってしまう．このため**補間**（interpolation）が必要になる．TREC では，例えば，再現率 0.1 の点での精度を補間する場合，0.1 よりも大きなすべての再現率に対応する精度の中での最大値を用いる [10]．例えば，$R = 3$ で，これらの適合文書の出力順位が 4，9，20 位ならば，それぞれの再現率は 0.33，0.67，1.0 で，精度は 0.25，0.22，0.15 である．TREC の補間の規則によれば，この場合，再現率 0.0 〜 0.3 における精度はいずれも 0.25，0.4 〜 0.6 は 0.22，0.7 〜 1.0 は 0.15 となる．この例から容易にわかるように，TREC の補間規則は，再現率の増加に対して，精度が単調に減少するように設定されている．

trec_eval では実際にこの 11 個の補間された精度が計算される．このデータを使って，再現率の 11 個の点を x 軸にとり，それぞれの補間された精度の値を y 軸としてプロットしたものが**再現率－精度グラフ**（または**再現率－精度曲線**）である．検索実験では，再現率の 11 個の点ごとに，L 個の検索課題の精度を平均してグラフを描き，手法（システム）間の比較をおこなうことが多い．当然，11 個の精度の点を結んだ曲線がグラフの右上に位置するほど，性能が高いことになる．

3 検索モデル

3.1 検索モデルの種類

文書を順位付けて出力する際には，検索質問に対する各文書の適合度をなるべく正確に推計しなければならない．この推計には**検索モデル**が重要になる．現在までに提案されている主な検索モデルには，

- ベクトル空間モデル（vector space model）
- 確率型モデル（probabilistic model）

[8] ちなみに，Microsoft Excel の「分析ツール」では,「t 検定：一対の標本による平均の検定」に相当する．

[9] 市販の表計算ソフトで簡単に計算できる．

- ファジイモデル（fuzzy model）
- 言語モデル（language model）

などがある．NTCIRでは，ベクトル空間モデルと確率モデルが頻繁に利用されている．特に，確率型モデルでは，Okapi方式（後述）がよく使われている．

各モデルとも，それぞれの理論に基づいて，最終的に各文書の得点（score）を計算する．この文書得点はRSV（retrieval status value）とも呼ばれる．例えば，ベクトル空間モデルでは，文書ベクトルと検索質問ベクトルとの類似度を文書得点として算出する．一方，確率型モデルでは，検索質問が与えられたときの各文書の適合確率を基本として文書得点が計算される．検索モデルに基づいて文書得点を計算し，その降順に文書番号を並べれば，trec_evalを使って，その性能を評価することが可能になる．

最終的に文書得点を算出する必要があるため，各モデルとも，語の出現に関する何らかの統計量を利用せざるを得ない．一般に，各モデルには次の3種類の統計量が明示的・暗黙的に含まれる．

- tf（term frequency）：1件の文書における，ある語の出現頻度
- idf (inverse document frequency)：ある語が出現する文書数の逆数
- dl (document length)：文書の長さ

tf は文書 d_i における語 t_j の出現回数 x_{ij} である．ここで文書総数を N，語の総数（異なり）を M とすれば，$i = 1, ..., N$ かつ $j = 1, ..., M$ である．検索質問に含まれる語が繰り返し出現する文書ほど，その検索質問に対して適合している可能性が高いと考えられるので，tfを，その語の重要性を示す**重み**（weight）として用いることができる．

しかし一方，例えば学術文献における「研究」「議論」などの非専門語は，その主題内容を表現しないにも関わらず，文書に頻繁に出現する可能性がある．したがって，これらの語を無視するか，あるいはその重みを下げなければならない．このためには，「数多くの文書に出現する語ほど非専門的である」と仮定して，その語がデータベース中で出現する文書数（document frequency）を使えばよい．具体的には，語 t_j が出現する文書数 n_j を計算して，その数が閾値を超えるような語を無視するか，あるいはその語のtfに n_j の逆数（すなわち *idf*）を掛けてその重みを下げる．このようなtfとidfを使った重み付けのシステムを *tf-idf* と呼ぶ．

最後に，*dl*（文書長）は，語を数多く含む文書，すなわち長い文書が，不当に有利になるのを防ぐための補正に使われる．検索モデルの多くは，以上のtf，idf，dlを何らかのかたちで含んでいる．最終的な文書得点の計算のために，これらの要因をいかに数学的に組み合わせるかは，各モデルの理論・考え方に依存する．

以下，各モデルにおける代表的な文書得点の計算式を紹介する．検索モデルの詳細な理論や式の導出方法については省略するので，それぞれの文献を参照してもらいたい．検索モデルについての日本語の教科書もいくつか出されている [3, 11, 12]．なお，ファジイモデルは最近の利用例が乏しいことから，ここでは割愛する（文献 [3] などを参照）．

3.2 ベクトル空間モデル

ベクトル空間モデルは，G.Saltonを中心とするグループが自動索引・検索システム *SMART* を開発していくなかで整備されてきた検索モデルである．各語の重みから構成されるベクトルとして文書と検索質問をそれぞれ表現し，2つのベクトルの成す角度の余弦によって類似度を計算する点に特徴がある．

重みの計算にはいくつかの種類があるが，例えば，文書 d_i における語 t_j の重み w_{ij} を

$$w_{ij} = \log x_{ij} + 1.0 \tag{5}$$

とし（x_{ij} は d_i での t_j の出現回数），検索質問 q における t_j の重み w_{qj} を

$$w_{qj} = (\log x_{qj} + 1.0) \left(\log \frac{N}{n_j} \right) \tag{6}$$

とする方法がある [13]．ここで，x_{qj} は検索質問における t_j の出現回数，N は文書総数，n_j は語 t_j が出現する文書数である．

これら語の総数（異なり）を M とすれば，重み w_{ij} と w_{qj} を使って、文書と検索質問とをそれぞれ M 次元ベクトル

$$\mathbf{d}_i = (w_{i1}, w_{i2}, ..., w_{iM}) \tag{7}$$

$$\mathbf{q} = (w_{q1}, w_{q2}, ..., w_{qM}) \tag{8}$$

で表現できる．検索質問 q に対する文書 d_i の得点 $s_q(d_i)$ は 2 つのベクトルの角度の余弦，すなわち，

$$s_q(d_i) = \frac{\langle \mathbf{d}_i, \mathbf{q} \rangle}{\|\mathbf{d}_i\|\|\mathbf{q}\|} \quad (9)$$

$$= \frac{\sum_{j=1}^{M} w_{ij} w_{qj}}{\sqrt{\sum_{j=1}^{M} w_{ij}^2}\sqrt{\sum_{j=1}^{M} w_{qj}^2}} \quad (10)$$

で計算される．

3.3 確率型モデル

3.3.1 Okapi

S.E. Robertson を中心に開発された *Okapi* と呼ばれる次世代検索システムにおいて使用されている確率型の検索モデルは，ベクトル空間モデルと同等，あるいはそれ以上の性能を示すことでよく知られている．このモデルは，原理的には，検索質問 q と文書ベクトル $\mathbf{d}_i$ が与えられときに，その文書が検索質問に適合している確率 $P(R|q, \mathbf{d}_i)$ を推計するものである（ここでは R は適合文献数ではなく，適合事象を示す）．1970 年代には，この推計のために，2 値変数に基づくパターン認識の方法を模倣したモデルが使われていたが [14]，1980 年代から 90 年代にかけて，tf を組み込むために 2-Poisson モデルを活用した拡張が試みられた [15]．この拡張によって得られる計算式は非常に複雑であり，実際的でないことから，大幅な近似を導入することによって式の簡素化が図られた [16]．現在多くのチームで活用されている計算式はその一種で，BM25 と呼ばれているものである．その式を以下に示す．

$$s_q(d_i) = \sum_{j=1}^{M} (\omega_{ij} \times x_{qj} \times \tau_j) \quad (11)$$

ここで，

$$\omega_{ij} = \frac{3.0 x_{ij}}{(0.5 + 1.5 l_i/\bar{l}) + x_{ij}} \quad (12)$$

$$\tau_j = \log \frac{N - n_j + 0.5}{n_j + 0.5} \quad (13)$$

である．なお，

$$l_i = \sum_{j=1}^{M} x_{ij} \quad (14)$$

は文書 d_i の長さであり，

$$\bar{l} = \frac{1}{N} \sum_{i=1}^{N} l_i \quad (15)$$

はデータベース全体での文書長の平均を意味する．BM25 は実際には，k_1，k_2，k_3，b の 4 つのパラメータを持つモデルであり，上の式は，$k_1 = 2.0$，$k_2 = 0.0$，$k_3 = \infty$，$b = 0.75$ に設定した場合に得られるものである [16]．

3.3.2 INQUERY

W.B.Croft を中心に開発された INQUERY は，**ベイズ型推論ネットワーク**に基づく検索システムである．このシステムでは，文書 d_i が与えられたときの検索質問 q の確信度 $B(q|d_i)$ によって文書の順位が決められる．具体的には，

$$B(q|d_i) = \frac{\sum_{j=1}^{M} w_{qj} B(t_j|d_i)}{\sum_{j=1}^{M} w_{qj}} \quad (16)$$

であり（他にも計算方法があることに注意），ここで，

$$B(t_j|d_i) = 0.4 + 0.6 \times \frac{x_{ij}}{x_{ij} + 0.5 + 1.5 l_i/\bar{l}} \times \frac{\log \frac{N+0.5}{n_j}}{\log N + 1} \quad (17)$$

のように設定される（この式は Okapi の方式とよく似ている）[17]．

3.3.3 ロジスティック回帰モデル

California 大学 Berkeley 校の W.S. Cooper を中心に開発された方法では，**ロジスティック回帰モデル**を使って，適合確率が推計される [18]．正確には，適合確率の対数オッズ比 $\log O(R|\mathbf{d}_i, q) = \log \frac{P(R|\mathbf{d}_i,q)}{P(\bar{R}|\mathbf{d}_i,q)}$ を回帰式によってまず推計し，変換

$$P(R|\mathbf{d}_i, q) = \frac{1}{1 + e^{-\log O(R|\mathbf{d}_i,q)}} \quad (18)$$

によって元に戻してこれを文書得点とする．具体的に $\log O(R|\mathbf{d}_i, q)$ を求めるための回帰式は，

$$Y = -3.51 + 37.4 X_1 + 0.33 X_2 - 0.1937 X_3 + 0.0929 X_4 \quad (19)$$

が使用されており，ここで，

$$X_1 = \frac{1}{\sqrt{m}+1}\sum_{j=1}^{m}\frac{x_{qj}}{l_q+35} \tag{20}$$

$$X_2 = \frac{1}{\sqrt{m}+1}\sum_{j=1}^{m}\log\frac{x_{ij}}{l_i+80} \tag{21}$$

$$X_3 = \frac{1}{\sqrt{m}+1}\sum_{j=1}^{m}\log\frac{\sum_{i=1}^{N}x_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{M}x_{ij}} \tag{22}$$

$$X_4 = m \tag{23}$$

である[10]．m は検索質問と文書とが共有する語の数（異なり）であり，添字 j はこれらの語の集合に対して，$j = 1, ..., m$ のように番号を付与したものとする．l_q は検索質問の長さであり，検索質問に含まれる延べ語数として定義される．X_3 における対数の分子はその語の出現延べ総数であり，分母はデータベースの長さとして解釈できる．結局，Berkeley によるモデルでは，以上の回帰式に基づいて，文書得点を $s_q(d_i) = 1/(1+e^{-Y})$ によって求めることになる．

3.4 言語モデル

ある言語についての語 t_j（$j = 1, ..., M$）の出現確率の分布 $p_1, p_2, ..., p_M$ を**言語モデル**（language model）という [19]．この確率分布を推定するためのテキストの集合（コーパス）は有限であるから，真の分布を求めることは容易ではない．この問題に対する工夫が数理言語学の分野を中心に研究されている．このモデルを情報検索に応用するには，例えば，各文書に対しての確率分布を考え，そこから無作為に生成されるイベントとして検索質問を捉えることにより，その生成確率の推計値を文書得点とすること [20] などが考えられる．このモデルに基づく文書得点の計算方法の例としては，

$$s_q(d_i) = \prod_{t_j \in Q}\left(a\frac{x_{ij}}{l_i} + (1-a)\frac{n_j}{N}\right)^{x_{qj}} \tag{24}$$

がある．ここで，Q は検索質問に含まれる語の集合，a はパラメータである．

[10] X_1 にのみ log が使われていないことについては深い理由はなく，どうやら経験上このように設定されたらしい．同様に，35 と 80 というパラメータも検索実験の結果から試行錯誤的に求められたものである

3.5 LSI

ベクトル空間モデルでは，索引語によって張られる空間中に文書と検索質問を置き，その中での両ベクトル間の類似度を計算しているが，各次元は直行したままである．これは，各索引語が独立して類似度の計算に寄与することを意味している．この独立性の仮定を緩める努力はいくつか試みられてきたが，その中でも *LSI*（Latent Semantic Indexing）は，**特異値分解**（singular value decomposition：SVD）を使ってベクトル空間の次元を縮約することにより，語間の依存性を類似度の計算に持ち込もうとする方法である [21, 22]．理論的には，この方法によって縮約された後の各次元は，関連する語によって表現される「概念」に相当することになる．

まず，語×文書の重み行列を $\mathbf{W} = (w_{ji})$，$j = 1, ..., M$，$i = 1, ..., N$ とおく．これを，$\mathbf{W} = \mathbf{TSU}^T$ のように特異値分解する．ここで $\mathbf{T}$ は，$M \times r$ の直交行列，$\mathbf{S}$ は $r \times r$ の対角行列，$\mathbf{U}$ は $N \times r$ の直交行列，r は行列 $\mathbf{W}$ のランク，T は転置を示す記号である．

次に，$\mathbf{S}$ の対角要素をその大きさの上位 r' 個だけ選び，それ以外を 0 とした行列 $\tilde{\mathbf{S}}$ を作る．そしてこれを使って，重み行列を $\mathbf{T}\tilde{\mathbf{S}}\mathbf{U}^T = \tilde{\mathbf{W}}$ のように再計算したとする．この $\tilde{\mathbf{W}}$ を使って文書間の類似度を求めれば，それは，重要な概念に対応する次元に縮約された空間中における類似度を計算していることになる．例えば，あらかじめ $\mathbf{W}$ を各文書のノルムで正規化しておけば，

$$\tilde{\mathbf{W}}^T\tilde{\mathbf{W}} = (\mathbf{U}\tilde{\mathbf{S}})(\mathbf{U}\tilde{\mathbf{S}})^T \tag{25}$$

は，次元縮約された空間中での余弦係数による類似度行列となる．

そこで，検索質問を仮想的に文書とみなし，正規化した質問ベクトルを縦ベクトルとして $\mathbf{W}$ の第 1 列に挿入して（したがってこれは $M \times (N+1)$ 行列になる），式（25）を計算すれば，検索質問に対するそれ自身および各文書との類似度が，行列 $\tilde{\mathbf{W}}^T\tilde{\mathbf{W}}$ の第 1 列（または第 1 行）に計算されていることになる．したがって，この類似度を文書得点として用いればよい．

3.6 質問拡張の技法

質問拡張（query expansion）とは，一般に，利用者によって作成された検索質問（または検索語の集合）に語を追加することによって，検索性能を向上させようとする工夫を指す．現実の状況において，図書館員や検索専門家が，シソーラスや辞書，あるいは自分の知識に基づいて，検索語の追加・削除をおこなうのは日常的な作業であり，これに類した自動的機能をシステムが備えることによって性能向上がもたらされる可能性がある．

このための方法としては，

- 機械可読型のシソーラスや辞書などの外部資源を使った自動拡張
- 適合フィードバックの活用

などがある．**適合フィードバック**（relevance feedback）とは，第 1 段階での検索結果に関する適合情報を利用者から返してもらい，それを活用して第 2 段階（あるいはそれ以降）の検索性能を向上させる工夫である．具体的には，まず，初期的な検索（initial search）を実行してその検索結果を利用者に提示する．そして，利用者に各文書の適合・不適合を判定してもらい，判定された文書中に含まれる語句を使って検索語の追加・削除あるいは重みの調整（re-weighting）をおこなう．

適合フィードバックの方法としては *Rocchio* **の方法**がよく知られている [23, 24]．この方法は基本的にはベクトル空間モデルに基づいており，検索質問ベクトル $\mathbf{q}$ を以下の式によって新しいベクトル $\tilde{\mathbf{q}}$ に変換し，第 2 次の検索を実行するものである．

$$\tilde{\mathbf{q}} = \alpha\mathbf{q} + \frac{\beta}{|\mathrm{R}|}\sum_{i:d_i\in\mathrm{R}}\mathbf{d}_i - \frac{\gamma}{|\bar{\mathrm{R}}|}\sum_{i:d_i\in\bar{\mathrm{R}}}\mathbf{d}_i \qquad (26)$$

ここで，R と $\bar{\mathrm{R}}$ はそれぞれ適合文書と不適合文書の集合，$|\mathrm{R}|$，$|\bar{\mathrm{R}}|$ はそれらに含まれる文書数である．また，α，β，γ はパラメータである．つまり，Rocchio の方法とは，適合文書中の語の重みの平均から不適合文書中の語の重みの平均を差し引いたものを，元の質問ベクトルに加えることにほかならない．パラメータはこの加算の重みであり，例えば，$\alpha = 8$，$\beta = 16$，$\gamma = 4$ のように設定される [25]．

Rocchio の方法を適用するには，利用者によって，適合文書と不適合文書とが与えられなければならない．このような実際の利用者を想定した対話的な環境における検索実験が試みられることもあるが，多くの検索実験の状況下では利用者不在であり，適合フィードバックを直接的使った質問拡張は不可能である．ひとつの方法は，第 1 次の検索で上位に位置づけられた文書を適合と仮定して，フィードバックの方法を適用することである．これは**擬似適合フィードバック**（pseudo relevance feedback）と呼ばれており，これを用いない単純な検索に比べて，一定の性能向上をもたらすことが検索実験の結果として経験的に知られている．

擬似適合フィードバックの具体的な方法にはさまざまなものがあるが，単純なものとしては，式 (26) において $\gamma = 0$ と仮定して，そのまま Rocchio の方法を適用する方法がある．ただし，実際には，適合文書と仮定した上位の文書中での重みが高い順に語を並べて，その上位の何語かのみを式 (26) に従って追加する場合が多い（例えば，上位 10 文書中の上位 100 語を追加，など）．このときの語の重みを TSV（term selection value）と呼ぶことがある．

より複雑な方法として，適合フィードバックの方法より忠実に「再現」し，適合文書と仮定された上位の文書以外の文書，すなわち「擬似不適合文書」中の語の重みをも考慮する場合がある．簡単な例を以下に示す [26]．ある語 t_j を固定し，それが出現する適合文書数および不適合文書数を計数する．すると，表 2 のような 2×2 分割表が得られることになる．ここで，N は文書総数，R は適合文書総数，n_j は語 t_j が出現する文書数，r_j は t_j が出現する文書のうちの適合文書数である．古典的な確率型モデル [14] に従えば，この表から，

$$\omega_j = \log\frac{r_j(N - R - n_j + r_j)}{(R - r_j)(n_j - r_j)} \qquad (27)$$

によって語の TSV を計算すればよい．一般に，表 2 のような分割表における表側と表頭の関連度を測定する指標にはさまざまなものがある．

表 2: ある語の出現文書に関する分割表

	適合	不適合	計
出現	r_j	$n_j - r_j$	n_j
非出現	$R - r_j$	$N - R - n_j + r_j$	$N - n_j$
計	R	$N - R$	N

NTCIR でも，多くのチームが擬似適合フィードバックを活用しており，様々な独自の工夫が加えら

れている（例えば，文献 [27, 28, 29] など）．一般に，データベース中での文書集合から語の共起情報に基づいて自動作成したシソーラスによる質問拡張は検索性能の改善をそれほどもたらさないといわれてきた [30, 31]．その理由のひとつは，共起情報に基づく場合，適合文書はその片方の語（元の検索質問に含まれるほうの語）によってすでに検索されている可能性が高く，それゆえ「新たな」適合文書の検出は期待しにくいというものである．また，共起情報を抽出するデータベースが，用語法の異なる様々な分野・領域の文書から構成されているとすれば，それに対する大域的な分析結果としての共起情報はむしろ誤った（関連のない）語の追加をまねく危険性もある．このようなことから，共起情報に基づくシソーラスに比べて，擬似適合フィードバックに基づく分析のほうが，自動的な質問拡張としては効果があると一般に考えられている．ただし，擬似適合フィードバックを現実に適用する場合には，かなりの計算量が必要であることや（単純計算で通常の 2 倍以上），平均的には性能は向上するものの，場合によっては検索性能の低下をまねくことなど，解決すべき問題は数多い．

4 実験用検索システムの構築

4.1 検索システムの構成と処理手順

検索実験によって検索手法やモデルの性能を試める場合には，利用者インタフェースはとりあえず不要であり，テストコレクションとして用意された検索課題のファイルを入力し，文書得点順に並べた文書番号のファイルを出力すれば十分である．NTCIR などの検索実験に参加する場合には，この出力ファイルを trec_eval に適用できるよう整えて，オーガナイザに提出すればよい．または，自分で trec_eval を実行する場合には，テストコレクションの適合判定結果ファイルから各文書の適合／不適合の情報を抽出して，出力ファイルに組み込めばよい．

具体的な手順は，例えば，次のようになる．

1. 文書ファイルの各レコードに対して次の処理をおこなう．
   (a) 必要なフィールドからテキストを抽出する
   (b) テキストから語を抽出し，その出現回数を計数して，索引ファイルに登録する
   (c) 文書長を計算し，文書長ファイルに登録する
2. 検索課題ファイルの各レコードに対して次の処理をおこなう．
   (a) 必要なフィールドからテキストを抽出する
   (b) テキストから語を抽出し，その出現回数を計数する
   (c) 索引ファイルと文書長ファイルを使って，各文書の得点を計算していく
   (d) 文書得点の降順に文書を並べ替え，結果を出力する
3. 出力ファイルを整えて，オーガナイザに提出する．または，適合判定結果ファイルから適合情報を組み込み，trec_eval を実行する．

4.2 索引作成

NTCIR をはじめとする最近の検索実験では，XML に準拠したタグを使って各レコードを記述しているので，文書レコードや検索課題レコードからフィールドを識別し，テキストを抽出することは難しくない．テキストを抽出したら，次にそこから語を識別する．特に，検索に用いるための語を文書レコードから抽出する作業は**索引作成**（indexing）と呼ばれ，抽出された語を**索引語**（index term）という．

英語の場合には，単語は空白で区切られているので，語の抽出は容易である．次に，そこから the, of などの機能語を取り除く．これらの語を**ストップワード**と呼ぶ（ストップワードのリストは文献 [32] を参照）．この作業の結果，残った語が索引語ということになる．

ただし，英語などの語尾変化をする言語の場合には，**語幹抽出**（stemming）を施す必要がある．語幹抽出をおこなうと，例えば，library や libraries はいずれも librar となり，語形が一致する．語幹抽出にはいくつかのアルゴリズムが開発されているが [33]，Porter のアルゴリズム [34] がコンパクトで使いやすい．

日本語の場合には，語の間に切れ目を置かない，いわゆる膠着語なので，まず語分割（word segmentation）が必要である．明らかに，英語の場合よりも索引作成はやっかいであるが，幸い，最近では，

「茶筌」[35] などの優れた形態素解析システムが手軽に利用できるようになっており，これらのツールを活用すればよい．

英語の場合にも日本語の場合にも，**複合語**（compound term）の識別が検索性能の向上に貢献することがある．よい複合語の辞書があれば，その見出し語とテキストとを付き合わせて切り出していけばよい．複合語を豊富に掲載した辞書がないときには，例えば日本語の場合,「助詞（またはひらがな）を間に挟まずに隣接している語は自動的に組み合わせる」などの発見的な（heuristic）規則を設定する方法もある．

また，語を明示的に識別せずに，いわゆる **Nグラム**（N-gram）としてテキストを分割する方法もよく用いられる．日本語のような膠着語の場合には，文字単位の N グラムを使うことになるが，一般には，**バイグラム**（bigram）がよく利用される．具体的には，例えば「日本国憲法」の場合には「日本」「本国」「国憲」「憲法」のように重複して 2 文字ずつ抽出していく（overlapped bigram）．バイグラムは，索引ファイル（後述）が大きくなってしまうなどの意味でコストが高いが，語を抽出した場合と比べて，同等またはそれ以上の性能を示すことがある．

4.3 索引ファイル

索引ファイル（index file）とは，索引語をキーとして，索引語に関する情報をすばやく引き出せるように構成されたファイルを意味し，伝統的には**転置ファイル**（inverted file）などとも呼ばれる．索引ファイルが持つ情報としては，

- 出現文書数：idf の計算などに使う
- 出現文書：文書番号またはポインタを記録
- 各文書中での出現頻度：すなわち tf
- 各文書中での出現位置の情報

などがある（出現位置情報は記録されないこともある）．

文書集合が大規模である場合，この索引ファイルもかなり大きくなる．そこで，使用するコンピュータの環境にもよるが，索引ファイルを主記憶にすべて抱え込むようなプログラムを書いてしまうと主記憶の領域が不足してしまう可能性がある．したがって，応答速度の悪化は避けられないが，ハードディスクのような外部記憶装置との併用を考える必要がある．

簡単なアルゴリズムは **B木**（B-tree）であろう．B 木の一部分だけを主記憶領域上にのせ，膨大な情報を持つ索引ファイルの本体はハードディスク上に置いておけば，主記憶領域が小さくとも，かなりの大きさの文書ファイルを処理できる[11]．具体的なアルゴリズムについては，文献 [36] などが参考になる．

その他，索引ファイルを作成せずに，大規模テキストに高速にアクセスするための方法として *suffix tree* や *signature file* などがあり [37]，suffix tree は NTCIR でも使用しているチームがいくつかある．

4.4 文書得点の計算

前節で説明したベクトル空間モデルや確率型モデルによる計算式は一見複雑に見えるが，実は検索質問ベクトルに対する線形の関数にすぎない．これは，文書得点を求める際に，検索語ごとに独立して計算し，後でそれらを単純加算すれば十分であることを意味している．この結果，プログラムはずいぶん容易になる．

例えば，Okapi における BM25 を使って文書得点を計算する場合を考える．ある検索課題を処理した結果，m 個の語が識別されたとする．この場合，まず 1 番目の語に対して索引ファイルを探索し，それが出現する文書のみに関して，それぞれ，式 (11) における $\omega_{ij} \times x_{qj} \times \tau_j$ を計算して，その値のみを記録しておく．索引ファイル中に各文書の tf の情報さえ持っていれば，ただちにこの値を計算することが可能であり（ただし，文書長 l_i の情報だけは文書長ファイルから引き出さなければならない），しかもこの値を計算したのちは，この式に含まれる情報は捨ててもかまわない．後は，この処理を m 回繰り返し，文書ごとに上の式の値を単純に累積していけば，最終的に文書得点が計算できる．

ほとんどの検索モデルは質問ベクトルに対して線形であり，この点，実装は容易である．

[11] 筆者（岸田）は，NTCIR-2 に参加したとき，主記憶領域 96MB のパソコンですべてを処理した経験を持つ．

5 NTCIR-3での検索実験(1):言語横断検索

5.1 言語横断検索の概要

言語横断検索（cross-lingual retrieval）とは，一般に，検索質問と検索対象文書とが異なる言語で書かれている場合の情報検索を指す．例えば，日本語の検索質問をシステムに投入して，英語の論文を収録したデータベースを検索する場合などである．このためには，基本的には，検索質問と文書集合のどちらかを翻訳して他方に合わせる必要がある．現実的には，文書集合は巨大であり，翻訳に手間がかかるし，翻訳に伴って転置索引ファイルの規模も大きくなってしまう．そのため，多くの場合，検索質問の翻訳が試みられる．もちろん，文書集合の翻訳を試みている研究は存在するし，検索質問と文書集合の両方を翻訳するいわばハイブリッドな方式も考案されている [38].

しかし，Webのサーチエンジンでの多くの検索がそうであるように，検索質問が数語の検索語の単なる羅列から構成されるならば，翻訳に必要な情報が不足し，語の曖昧性（ambiguity）を解消することが難しくなる．例えば,“ Mercury ”という英単語が何の文脈も与えられずに単独で入力されたとき，これを神話上の人物である「マーキュリー」と解釈すべきか，あるいは惑星の「水星」と解釈すべきか，化学物質の「水銀」と解釈すべきか，決定しきれない．このための曖昧性解消（disambiguity）の方法についての研究が進められている [39, 40].

自動的な言語横断検索の歴史は比較的新しく，1996年におこなわれたACM-SIGIR Workshop on Multilingual Informationにおいて言語横断検索の概念が明確にされ，1997年のAAAI Spring Symposium on Cross-Language Text and Speech Retrievalにおいて，さまざまな手法が報告され，方法論的な基盤が確立されたようである [41, 42]．ほぼ同時期に，TRECの第6回目（TREC-6，1997年）において，言語横断検索が研究課題として取り上げられ，それ以降，英語とスペイン語，フランス語，ドイツ語などの欧州諸言語間の言語横断検索が精力的に研究された．この言語横断検索は，従来の情報検索分野の研究者だけでなく，自然言語処理研究者や人工知能研究者の関心も広く集め，現在では，非常に活発な研究領域となっている．

NTCIRではすでに述べたように，第1回目に英語と日本語の言語横断検索が取り上げられ，その後，中国語・韓国語といった東アジア言語に範囲を拡大している．NTCIRでの成果は，すでに学術論文としていくつか発表されている（例えば [43]）[12]．また，NTCIR-1で使用された文書データベース（国立情報学研究所がNACSIS-IRで提供している『学会発表データベース』や『科学研究費補助金研究成果概要データベース』が基になったもの）に含まれる多くの発表・報告は和文抄録と英文抄録，和文著者キーワードと英文著者キーワードとを持っており，これを対訳コーパス（後述）として活用した研究例もある（例えば [40]）.

また，ヨーロッパでは，TRECやNTCIRと同様なプロジェクトであるCLEF （Cross-Language Evaluation Forum）[13] が活動中であり，さらに米国ではTIDES（Translingual Information Detection, Extraction and Summarization Program）[14] プロジェクトが実行されている．これは言語横断検索だけではなく，より幅広く，言語横断的な情報抽出（information extraction）や要約（summarization）の問題にも取り組もうというプロジェクトである．

5.2 言語横断検索の技術

検索質問または文書のテキストを翻訳する方法としては，

- 対訳辞書やシソーラスの利用
- 対訳コーパスの利用
- 機械翻訳システムの利用

が代表的である [44, 45]（なお，上で紹介したLSIを使って，直接的な翻訳をおこなわずに，言語横断検索を実現しようとする試み [46] もある．この方法は対訳コーパスを使った手法として分類されることがある）.

まず，**対訳辞書**とは，例えば，英和辞典，和英辞典のような辞書を機械可読形式にしたもの（Machine Readable Dictionary: MRD）である．対訳辞書を使う場合，処理手順は次のようになる．

[12] http://research.nii.ac.jp/ntcir/paper1-ja.html
[13] http://www.iei.pi.cnr.it/DELOS/CLEF/
[14] http://www.darpa.mil/ito/research/tides/

1. 検索質問に対して索引作成を実行し，語を抽出する
2. 抽出された語を，対訳辞書を使って，文書集合で使用されている言語の語に変換する
3. 変換された語を使って検索を実行する

ここで問題になるのが語の多義性であり，つまり，対訳辞書中の 1 つの見出し語に対して，複数の訳語が存在する可能性がある．この解決方法としては，(a) すべての訳語を採用する，(b) 先頭の訳語を採用する，(c) 品詞情報を活用する（例：名詞を動詞よりも優先する）などがある [42]．しかし，上の例の“ Mercury ”のように，意味の異なる複数の訳語を持つ場合には，何らかの**曖昧性解消**（disambiguation）を施さなければ，検索ノイズを出力する原因となってしまう．例えば，もし，検索語が複数与えられれば，それぞれの訳語について，文書集合中の共起頻度を計算すれば，ある程度の曖昧性解消が可能となる（例えば，検索語が Mercury と planet であるならば，文書集合中で「惑星」と共起しやすい「水星」を訳語として採用すればよい）．

また，対訳辞書中に語が登録されていない場合にも問題が生じる．特に，専門用語を幅広く網羅した対訳辞書は利用できないことが多い．この解決方法が**対訳コーパス**（parallel corpora または aligned corpora）の利用である．コーパス（corpus）とは，ある特定主題に関する資料の集まりを意味し，同等の内容を持つ 2 言語以上の文書の組みから成るものを対訳コーパスという [47]．

例えば，次の 2 文がそれぞれ他方の翻訳であるとする．

(1)The retrieval system allows us to search a bibliographic database

(2) この検索システムによって書誌データベースの探索が可能である

このような文を並置文（aligned sentences）と呼ぶ．ここでこれらの文がそれぞれ次のように語分割されたとする（実際には，英語の文に対しては語幹抽出を施すべきである）．

(1')the / retrieval / system / allows / us / to / search / a / bibliographic / database

(2') この / 検索 / システム / によって / 書誌 / データベース / の / 探索 / が / 可能 / である

この場合，上記の 10 個の英単語と 11 個の日本語の語のすべての組み合わせ（10×11=110 組）に対して，これらが 1 つの**並置**（alignment）に共起したと見なして，共起頻度 1 と計数できる（実際には，ストップワードや助詞等をはずせば，組み合わせの数はかなり減る）．

対訳コーパスによってこのような並置文が数多く得られれば，ある特定の英単語（例：database）と日本語の語（例：データベース）に対して，次のような統計量を計算することが可能になる．

(a) 両者が共起する並置文の数：n_{jk}

(b)「database」を含む並置文の数：n_j

(c)「データベース」を含む並置文の数：n_k

そして，これらの統計量から，両者の関連度 r_{jk} を，例えば，

$$r_{jk} = \log_2 \frac{N \times n_{jk}}{n_j \times n_k} \tag{28}$$

のように算出する．ここで N は並置文の総数である．英語から日本語への翻訳の場合には，各英単語に対して，関連度の高い日本語の語をいくつか記録しておき，それを一種の対訳辞書として使用することが可能になる．なお，式 (28) は相互情報量であるが，この関連度の計算にはさまざまな方法が研究されている．また，以上の方法は完全に統計的な方法であるが，文法が比較的類似した言語間では，対訳辞書を援用したアルゴリズムによって，対応する語を探索する方法も研究されている [48]．

最後に，機械翻訳システムを利用する場合には，基本的には，検索質問を機械翻訳システムに入力し，その出力結果をそのまま検索システムに投入することになる．検索質問が文章として表現されておらず，単なる検索語の集合である場合，この方法は効力を発揮しない．また，これまでの実験結果では，対訳辞書などの方法に比べて，機械翻訳システムはそれほどの性能を実現できていないようである（例えば [39] など）．

以上の諸手法を適用する場合，質問拡張を併用すると，一般に検索性能がさらに向上する．質問拡張の方法についてはすでに説明したが，言語横断検索の場合には，(a) 翻訳前と (b) 翻訳後の 2 通りの質問拡張が考えられる．例えば，検索質問を英語から日本語に翻訳する場合，まず，英語の検索語を拡張しておいて，それらに対して翻訳を実行する（翻訳前の拡張）．次に，翻訳された日本語の検索語に対して拡張をおこなう（翻訳後の拡張）．言語横断検索の

場合，質問拡張は曖昧性解消の手法のひとつとして機能することがある．なお，質問拡張のために，擬似適合フィードバックを使う場合には，翻訳前の元の言語に対応する文書集合が必要になる（翻訳後の質問拡張は，通常の質問拡張と同じであり，当然のことながら，検索対象となる文書集合を使用する）．

文書集合が単一の言語でなく，複数の言語の文書から構成されている場合には，言語横断検索の状況はより複雑になる．例えば，英語の検索質問に対して，日本語・中国語・英語から成るデータベースを検索する場合がこれに相当する．このための方法としては，一般に，

- 出力文書リストの併合：検索質問を対応する言語に翻訳し，それらを使ってそれぞれ単言語検索を実行して，最後に，それぞれの文書リストを併合する
- 検索質問の併合：検索質問を対応する言語に翻訳したのち，それらを併合して，複数の言語から成る検索質問に対して，多言語から構成される文書集合を検索する

がある．

出力された文書リストを併合するためには，

1. 各文書リスト中の得点をそのまま使う
2. 各文書リストの得点に何らかの変換を施す
3. 各文書リストの順位のみに着目して，交互に取り混ぜる（round-robin merging）

などの方法がある．1. の場合には，各文書リストの得点が比較可能であると仮定することになる．使用する検索モデルや文書集合の状況により，この仮定を置くことができない場合には，2. の方法を採らざるを得ない．これに関しては，まだ確立された方法はないが，複数のサーチエンジンあるいは複数のデータベースからの結果リストの併合に関する研究は数多くなされている [49]．例えば，ある 1 つの文書リストの得点を標準化する方法として，各文書の得点を変数 s，リスト中でのその最大値を s_{max}，最小値を s_{min} として，

$$\tilde{s} = (s - s_{min})/(s_{max} - s_{min}) \qquad (29)$$

を計算する方法はよく利用されている．この変換によって，文書得点は $0 \leq \tilde{s} \leq 1$ に標準化される．もっとも，式 (29) を適用して文書リストを併合しても，検索性能が向上するという保証はない．最後に，3. の方法は，文書得点を考慮せずに，まず各リストの 1 位だけを抜き取って順に並べ，その操作を 2 位，3 位，... と続けていくものである．各文書集合に含まれる適合文書数が異なれば，この方法では当然，問題が生じることになる．

一方，検索質問を併合する方法では，各言語別の文書総数が異なる場合に問題が生じる可能性がある．例えば，検索モデル中の idf に文書総数 N が含まれている場合，各言語別の文書集合を併合することによってこの N が大きくなる反面，各語の出現文書数は不変である．この結果，文書総数の少ない言語で書かれた語の idf が大きくなり，文書得点が不当に大きくなる可能性がある．

5.3 言語横断検索研究の課題

もし仮に，ある検索質問が日本語と英語とで表現され，しかもそれらが意味的に完全に同等であるならば，日本語の文書集合に対して，英語の検索質問から言語横断検索を実行した結果は，日本語の検索質問を使って普通に検索した結果よりも劣ることになると想定される．対訳辞書を使った場合には，言語横断検索の性能は通常の検索の 40 〜 60 %程度という報告がある [45]．この割合を 100 %に近づけることが，言語横断検索のひとつの目標である．このためには，曖昧性の解消，専門用語や複合語の翻訳の方法などを改善していくことが重要である．

また，中間言語を応用した多段階の言語横断検索も重要な課題である．例えば，日本語から中国語への翻訳しようとした際に，よい対訳辞書が利用できなかったとする．その場合，日本語から英語に一旦翻訳し，英語をさらに中国語に翻訳することが考えられる（各言語とも，英語に対する対訳辞書は比較的充実しているので，この方法が可能な状況はかなり多いと考えられる）．このような中間言語（あるいは pivot langauge）を利用した方法は，transitive translation approach と呼ばれることがあり [44]，その性能を，直接的な言語横断検索（ここでの例では日本語から中国語への翻訳）に近づけることは重要な研究課題のひとつである．

最後に，上で指摘した，複数の言語から構成され

る文書集合に対する言語横断検索については，効果的な方法に関して合意が得られておらず，今後とも研究を進めていく必要がある．

5.4 NTCIR-3での言語横断検索タスク

NTCIR-3 の言語横断検索タスクでは，日本語・中国語・韓国語・英語から構成される多言語データベースが用意されている（内訳は表 3 参照）．今回の検索対象は新聞記事であり，日本語と中国語に関しては，日本および台湾における 1998 年から 99 年にかけての記事がそれぞれ約 25 万件，合計で約 50 万件がデータベースに含まれている．一方，残念ながら，韓国語と英語のデータの規模は小さく，しかも韓国語の場合には年代がずれている（1994 年）．

表 3: 言語横断タスクにおける文書集合

言語	文書データベース	文書件数
日本語	毎日新聞 (1998-1999)（日本）	220,078
中国語	CIRB011 (1998-1999)（台湾）	13,217
	CIRB020 (1998-1999)（台湾）	249,508
	*CIRB: Chinese IR Benchmark	
韓国語	Korea Economic Daily (1994)（韓国）	66,146
英語	毎日デイリーニュース (1998-1999)（日本）	12,723
	EIRB010 (1998-1999)（台湾）	10,204
	*EIRB: English IR Benchmark	
	（Taiwan News (1998-1999) および	(7,489)
	Chinatimes English News (1998-1999) ）	(2,715)

これらの文書集合に対して，今回は次のようなトラック（個別的な研究課題）が設定されている．

- 多言語横断検索 (MLIR)：韓国語文書データの年代が他の言語のそれと異なるため，検索対象文書は中国語，日本語，英語の 3 言語，検索課題は，韓国語，中国語，日本語，韓国語，英語のいずれか 1 つ
- 2 言語横断検索 (BLIR)：文書集合の言語は単一で，検索課題の言語と検索対象文書の言語が異なる．検索課題は 4 言語，文書集合は英語を除いた 3 言語
- 単言語検索 (SLIR)：検索課題と検索対象文書の言語は単一かつ同一．ただし，英語文書データは文書数が少ないので，検索課題と文書の両方が英語という組合せは除く

このうち単言語検索は，検索質問と文書とが同一言語であるという点で，従来型の情報検索に相当する．

6 NTCIR-3での検索実験 (2)：特許検索

6.1 特許検索についての研究動向

特許検索については，各社共に研究開発が行われており有料サービスも多い．特許庁自身も無料で特許検索サービスを提供している[15]．一方，情報検索の基礎研究においては，特許検索が扱われることはあまりなかった．従来の情報検索は，とちらかというとジャンルに依存しない一般的な枠組みを目指していたからである．また，特許検索に関するテストコレクションが無かったことも原因の 1 つである．

それでも最近は，知的財産権の重要性が増してきたことから，特許検索や特許分類に関する研究発表が増えてきた．米国の TREC においても，特許を扱うタスクは行われていないものの，検索対象の一部に特許文献が含まれている．

特許に特化した学術会議としては，SIGIR2000 で開催された「特許検索に関するワークショップ (ACM-SIGIR Workshop on Patent Retrieval)」が初めての会議である[16]．このワークショップでは 9 件の発表とパネルディスカッションが行われ，様々な角度から特許検索の現状，将来について議論された．

上記の状況の中で，特許検索に関する研究を促進するためには特許に関するテストコレクションを作成することが必須と考え [50]，NTCIR-3 において特許検索タスクを実施した．世界的に見ても，広く入手可能な特許検索用テストコレクションはまだない．

[15] 特許庁特許電子図書館
http://www.ipdl.jpo.go.jp/homepg.ipdl

[16] ACM-SIGIR Workshop on Patent Retrieval
http://research.nii.ac.jp/ntcir/sigir2000ws/

6.2 特許検索の特徴と技術

特許文献は多くの点で，新聞や学術論文等の文書と異なっている．以下，特許文献 (明細書や公報と呼ばれるもの) が持つ主な特徴を列挙する．

- 構造を持った文書である．特許文献は書誌情報に加え,「請求項」「従来の技術」「発明が解決しようとする課題」「発明の実施の形態」等から構成されている．この中でも特に「請求項」は発明の範囲を定める重要な構成要素である．
- 「請求項」は慣例的に一文で記述されるため，文が長くなり非専門家には読みにくい．
- 「請求項」では発明の適用範囲を広げるために一般的な用語を用いることが多い．例えば,「ゴム」と書かずに「弾性体」と書く．
- 発明であるため新語や専門用語が多い．
- 長さにばらつきがある．長いものは非常に長い．
- IPC(国際特許分類), FI, F タームなどの分類コードが付与している．

特許検索に関しては，特許出願のフェーズに即した検索モデルが必要となる．例えば，製品開発の初期段階では，技術動向調査という観点で特許を検索することが多いため，新聞検索等で使われる一般的な検索モデルでも十分であるが，出願前の先行例調査では，漏れのない検索が必須となる．特許を活用したり製品開発の重点項目を決めたりするには，検索に加え，関連特許間の関係を一覧する特許マップの作成も有効である．

6.3 NTCIR-3 での特許検索タスク

NTCIR-3 では，特許に関する初めての評価会議ということもあり，従来からある検索系タスクと親和性の高い技術動向調査に焦点を絞った．

特許データベースとして，(株) パトリスから提供を受けた以下のデータを配布している．

1. 公開特許公報全文データ (98,99): 出願から 18ヶ月たって公開される特許全文データ．約 70 万文書，18G バイトから成る．
2. JAPIO 出願抄録データ (95-99): JAPIO((財) 日本特許情報機構) により作成された特許抄録データ．出願人要約の約半数を JAPIO が修正したもの．長さの統制 (400 字程度)，専門用語の統制が主な修正点である．約 170 万文書，1.8G バイトから成る．
3. 日本国英語特許出願抄録データ (95-99) : JAPIO 出願抄録データを英訳した英文抄録データ．約 170 万文書，2.7G バイトから成る．

設定した検索タスクは，製品開発前に行う技術動向調査である．例えば，会社のマネージャーが製品発表／開発に関する新聞記事に興味をもち，その新聞記事に関連する特許を検索する (検索を指示する) といった状況を想定している．よって，新聞から特許を探すデータベース横断検索となっている点も特徴である．

作成した検索課題には，新聞記事に加え，課題作成者が付与した“ description ”や“ narrative ”, 新聞記事の情報源となったと思われる特許の公開番号，等も含まれている．更に．日本語版の検索課題に加え，英語版，韓国語版，中国語版の検索課題も用意したため，日本語以外の言語から日本語特許を探す言語横断特許検索も評価可能である．

検索課題の作成および検索結果の判定はすべて，日本知的財産協会と共同で行った．検索結果の判定については，プーリングに先立って事前検索をおこなって，できるだけ多くの適合特許を集めてもらった．よって特許専門家による検索と参加システムによる検索との比較ができるようになっている．配布するテストコレクションから，事前検索による検索結果のみを抽出することも可能である．

検索結果の評価には trec_eval を使い，再現率－精度グラフ，平均精度，R 精度で評価した．タスクの目的が技術動向調査であるため，漏れのない検索結果を重視するといった特許特有の評価基準は考慮しなかった．また，今回は請求項を特別扱いすることをせず，特許を科学技術論文として扱った．

7 NTCIR-3 での検索実験 (3)：Web 文書検索

7.1 Web 文書検索の特徴と技術

World Wide Web による情報提供が増加して情報流通の基盤となるに従い，Web サーチエンジンの重要性も高まってきた．Web 上の情報の単位となるのが Web 文書であり，主にこれを対象にした情報検索が Web 検索と呼ばれる．

従来の情報検索が扱ってきた新聞記事，特許，論文などと異なり，Web 文書には検索の観点から見ると次のような特徴がある [51].

1. 作成者，作成目的の多様性: 情報の信頼性，記述の専門性，想定読者など
2. ジャンルの多様性: 論文，カタログ，議事録などから個人のプロフィール，日記などまでが区別なく混在
3. 表現の多様性: タグを用いたレイアウトや構造化，フレーム，表や画像などの視覚効果
4. 情報の粒度: 複数文書から構成される情報，複数情報が記載された文書
5. リンクによる参照: 参照，被参照の統計的，内容的情報の活用が可能
6. 変化の速度: 文書の追加，削除，更新が常時発生

また，Web 検索において効果的な検索を難しくしている以下のような要因もある．

1. 検索に関する情報量の不足: 検索条件として与えられる情報が少ない，利用者属性が特定できない，セッションがない
2. 利用者，利用目的の多様性: 情報収集，特定サイトの閲覧，サービスの利用など
3. Web 空間の規模とボーダーレス性: 網羅的収集が不可能，収集戦略が検索目的に依存，不特定の多言語など

Web 検索システムではこれらの特徴に対応するために，以下に例を示すようなさまざまな手法が提案され，サーチエンジンにおいて実現されているものもあるが，研究課題もまた多く残されている．

1. テキスト処理を中心とした伝統的な情報検索技術
2. タグ構造などを利用した重要部分の抽出
3. リンク参照の統計的処理によるページ重要度計算と，その検索結果ランキングおよびページ収集戦略への利用
4. 内容やリンク，URL に基づくクラスタリング
5. アンカーテキストを活用した被参照ページの索引補強
6. アクセスログを用いたサイト，Web 文書，アクセス傾向，利用者属性などの分析
7. 対話処理による効果的な検索条件入力

しかし，Web 検索技術を実際に研究の対象としようとすると，さまざまな難しい条件が存在する．まず，Web 検索システム全体の検索性能を左右する要因が，検索手法以外にも収集ページ数，ハードウェア性能，ページデザインなど多く存在するため，検索手法単独の評価や他の検索手法との比較評価を客観的に行うことが困難である．また，検索対象である Web 空間が巨大であるため，現実的な規模での実験には大量の計算機資源が必要となり，実用化まで見据えた本格的な研究が行いにくい．計算機資源があったとしても，実験のためには Web 検索システム全体を用意する必要があるため，初期開発量が大きい．さらに，Web 文書の収集には著作権等の適切な処理が必要でり，Web サーバの迷惑にならないような Web ページ収集上の配慮も要することから，研究以外での労力が過大になりがちである．このようなことから，新規に Web 検索技術の研究を開始するためのハードルはかなり高い． 最後に，利用者の検索要求と検索結果に対する評価のモデルが十分に研究されていないため，客観的評価のための基準が欠如している．例えば，(a) 検索条件は既存のサーチエンジンの利用実態を反映させるか，利用者の情報ニーズをより良く表現できるものにするか，また (b) 検索結果の適合判定ではテキストだけを見るか画像なども考慮するか，(c) リンク先のページも見るか，見るとしたらどこまで見るか，(d) 類似ページやリンクされたページをどうカウントするか，(e) 内容の信憑性や重要性をどのように評価するか，というように，従来の情報検索の評価尺度では評価できない事項が多くある．

7.2 NTCIR-3 での Web 検索タスク

Web 検索技術の研究において前節で述べたような全ての側面を同時に扱うことは困難であるので，テストコレクションでは Web 検索の多様な側面を切り出し，境界条件を固定して明確に定義することにより，実験と評価を可能とする研究基盤を提供することが重要な役割となる．同時に，Web 検索をとりまく諸要素に関しては未解明の部分が多いため，実験と評価の結果を現実とつき合わせることによりさまざまな特性を明らかにするためのテストベッドとしての役割も期待される [51]． Web 検索は第 3 回 NTCIR ワークショップで始めて取り上げるタスクであり経験も知識も十分でないため，基本的に従来の情報検索システム評価用テストコレクションの考え方を継承しつつ，可能な範囲で Web 文書や Web 検索の特徴に対応する，という基本方針で実施した．具体的には以下の通りである．

(a) 文書データにはオーガナイザが実際に収集した Web ページからなる共通の Web 文書データを用いることにより，Web 文書の収集や時間依存性に関する要因を排除する．主として JP ドメインから収集した．文書集合の規模は，参加者が扱いやすい規模 (10GB 程度) と，ある程度現実を意識した規模 (100GB 程度) を用いた．著作権の扱いやデータ容量に配慮し，文書データは国立情報学研究所に設置したオープンラボ内でのみ処理を行うこととした．

(b) 検索課題は課題作成者が持つ検索要求を一定の書式に書き下したものを用いることにより，検索条件入力方式に関する要因を排除した．特に，検索課題におけるタイトル ⟨TITLE⟩，主要適合文書 ⟨RDOC⟩ の定義，検索要求説明 ⟨NARR⟩ の書式について，独自な観点を採り入れた．

(c) 出力文書の適合判定は，検索課題の話題と出力文書中のテキストと間の適合性を基本とするものの，一定の条件を満たすリンク先の文書も判定対象に含める[17] ，適合性を多段階で判定する，文書の信頼性や文書中に占める関連部分の割合などの補助的な判定も行う，というように Web 文書の特徴を配慮した．

(d) システム評価については，利用目的として情報収集を念頭に置いて広く情報を収集するような場合に相当するサーベイ検索と，特定の情報を探すような場合に相当するターゲット検索のそれぞれに適した評価手法を用いた．個別文書の適合判定結果を用いて評価するものの，(c) で述べた通り，適合判定でリンク関係を考慮することによって，リンクを考慮したシステム評価となっている．

以下においては，タスク設計および評価尺度について具体的に説明する．

7.2.1 タスク設計

7.2(a) に述べた 100GB，10GB の文書データそれぞれに対して，以下のタスクに関する評価方式を設計した[18] [52, 53].

(A) サーベイ検索: 利用目的として情報収集を念頭に置き，広く情報を収集するような場合に相当する．検索結果上位 100 件程度以上に対する適合判定に基づいた，伝統的な精度および再現率に基づく評価尺度，後述の DCG（discounted cumulative gain）と呼ばれる評価尺度を用いる．

(A1) 検索課題検索: 検索課題から抽出した語を用いた検索 (以下，自動システム)，検索課題を参照しつつ対話的に実行する検索 (以下，対話型システム) について評価する．これは従来の Ad-hoc 検索タスクと類似したタスクと言える [54, 55, 56]．自動システムの場合，⟨TITLE⟩ を使用した検索と，⟨DESC⟩ を使用した検索を必須とし，それ以外の任意の部分を使用した検索を行なってもよいこととした．

(A2) 類書検索: 少数の適合文書に基づいた検索手法の評価を目指す．具体的には，(i) ⟨RDOC⟩ の先頭文書のみを必ず使用し，⟨TITLE⟩ の使用を妨げない場合と， (ii) ⟨RDOC⟩ の先頭文書を必ず使用し， ⟨RDOC⟩ における残りの文書および ⟨TITLE⟩ の使用を妨げない場合を想定し，(i) を必須とする．ただし，適合判定は検索課題の話題との関連性に基づい

[17] One-click-distance Document Model と呼ぶ．ただし，リンク先文書が全参加者の出力文書からなるプールに含まれる場合にのみリンク先を判定対象に含めるという制約を課した．これとは別途に，リンク先を考慮しないという条件（Page-unit document model）でも適合判定を行なった．

[18] http://research.nii.ac.jp/ntcir/workshop/web/

て実施する．

(B) ターゲット検索: 特定の情報を探すような場合に相当し，検索結果上位における精度が重視されるようなタスクを評価する．検索課題検索と同様，自動システムと対話型システムを受け付けるが，評価尺度については検索結果上位10件程度に対する精度に基づく尺度のほか，DCG，最初に検索された適合文書の順位の逆数に基づく尺度（後述の Mean Weighted Reciprocal Rank）を使用する．

(C) 自由課題: 自由に研究課題を設定し，研究を進める．(C1) 分類出力タスク，(C2) 音声入力タスクが企画されたが，結果としては，(C1) は実行結果の提出件数が 0，(C2) は企画者 1 チームのみが実行結果を提出するにとどまった．

各タスクの詳細などに関しては Web 検索タスクのホームページ（脚注 18）を参照されたい．

7.2.2 評価尺度

精度と再現率

サーベイ検索タスクでは,（非補間）平均精度，R 精度，上位 5，10，15，20，30，100 件における精度，再現率-精度グラフを用いた．これらは，trec_eval を使用して算出した．

一方，ターゲット検索タスクでは，上位 5，10，15，20 件における精度を使用した．

Discounted Cumulative Gain

以下で定義される Discounted Cumulative Gain（DCG）[57] を算出し，サーベイ検索タスクおよびターゲット検索タスクの評価に用いた．

$$\mathrm{dcg}(i) = \begin{cases} \mathrm{g}(1) & \text{if } i = 1 \\ \mathrm{dcg}(i-1) + \mathrm{g}(i)/\log_b(i) & \text{otherwise} \end{cases}, \quad (30)$$

$$\mathrm{g}(i) = \begin{cases} h & \text{if } \mathrm{d}(i) \in H \\ a & \text{if } \mathrm{d}(i) \in A \\ b & \text{if } \mathrm{d}(i) \in B \end{cases} \quad (31)$$

ここで，$\mathrm{d}(i)$ は i 番目の文書を，H，A，B はそれぞれ] 高適合，適合，部分的適合文書の集合を示す．2 種類の適合レベルを想定して，(31) 式における $g(i)$ の値を定めた．

適合レベル 1: $(h, a, b) = (3, 2, 0)$,

適合レベル 2: $(h, a, b) = (3, 2, 1)$.

(30) 式における対数関数の底は $b = 2$ とした．

サーベイ検索タスクでは上位 1,000 件まで，ターゲット検索タスクでは上位 20 件までの文書について，検索課題ごとに上記の DCG を計算し，平均値を算出した．

Mean Weighted Reciprocal Rank

Mean reciprocal rank（MRR） [58] は，質問応答システムの評価に用いられることが多く，質問ごとに最初に出現した正解の順位の逆数を求め，それらを全質問にわたって平均することで定義される．

NTCIR-3 Web 検索タスクでは，MRR の考え方をターゲット検索タスクの評価に適用した．次の wrr を全検索課題にわたって平均することで Mean weighted reciprocal rank（MWRR）を定義する．

$$\mathrm{wrr}(m) = \max\left(\mathrm{r}(m)\right), \quad (32)$$

$$\mathrm{r}(m) = \begin{cases} \delta_h / (i - 1/\beta_h) & \text{if } (\mathrm{d}(i) \in H \ \wedge \ 1 \le i \le m) \\ \delta_a / (i - 1/\beta_a) & \text{if } (\mathrm{d}(i) \in A \ \wedge \ 1 \le i \le m) \\ \delta_b / (i - 1/\beta_b) & \text{if } (\mathrm{d}(i) \in B \ \wedge \ 1 \le i \le m) \\ 0 & \text{otherwise} \end{cases} \quad (33)$$

ここで，m は実行結果において評価に考慮する最大の順位を示し，重み係数はそれぞれ $\delta_h \in \{1, 0\}$, $\delta_a \in \{1, 0\}$, $\delta_b \in \{1, 0\}$, $\beta_b \ge \beta_a \ge \beta_h > 1$ を満たすものとする．ところで，β_x の値が充分に大きいとき，(33) 式において，$(-1/\beta_x)$ $(x \in \{h, a, b\})$ の項は省略できる．

MWRR を算出する際には，m を 5，10，15，20 のそれぞれに設定した．また，次のように 2 種類の適合レベルを想定して δ_x を定め，簡単のため，β_x の値は充分大きいものと仮定した．

Level 1: $(\delta_h, \delta_a, \delta_b) = (1, 1, 0)$, $(\beta_h, \beta_a, \beta_b) = (\infty, \infty, \infty)$

Level 2: $(\delta_h, \delta_a, \delta_b) = (1, 1, 1)$, $(\beta_h, \beta_a, \beta_b) = (\infty, \infty, \infty)$

8 研究のための資源

情報検索研究のためにインターネット上で利用可能は情報資源については，NTCIR のホームページ

にリンク集がある[19]．これによって，NTCIR 以外のテストコレクション，辞書・機械翻訳システム，形態素解析システムなどが利用できる．また，文献 [45] には，言語横断検索のためのさまざまな資源が掲載されている。さらには，情報処理学会の機関誌『情報処理』の特集記事中に自然言語処理関連ツールの紹介がある [59]．

9 おわりに

本稿では，NTCIR ワークショップを中心に，検索実験の方法やそこでの研究課題，さらにはその実際的な問題点などを解説した．より詳しくは以下に掲げた文献を参照していただきたい．

参考文献

[1] 関根聡, 井佐原均, and 栗山和子. 日本におけるテストコレクションと評価の動向. **情報処理**, 41(8):902–905, 2000.

[2] 神門典子. NTCIR とその背景：情報アクセス技術の評価ワークショップとテストコレクション. **人工知能学会誌**, 17(4):296–300, 2002.

[3] 岸田和明. **情報検索の理論と技術**. 勁草書房, 1998.

[4] 栗山和子, 神門典子, 野末俊比古, and 大山敬三. 大規模テストコレクション構築のためのプーリングについて：NTCIR-1 の予備テストの分析. In **情報処理学会情報学基礎研究会** *54-4*, pages 25–32, 1999.

[5] 栗山和子, 吉岡真治, and 神門典子. 大規模テストコレクション NTCIR-2 の構築：言語横断的プーリングの評価への影響. **情報処理学会情報学基礎研究会**, (2001-FI-63), 2001.

[6] 神門典子, 栗山和子, and 吉岡真治. 多階段レレバンス判定による評価：平均可能な単一指標の検討. **情報処理学会情報学基礎研究会**, (2001-FI-63), 2001.

[7] C. Buckley and E. Voorhees. Evaluating measure stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, 2000.

[8] 岸田和明. 検索実験における評価指標としての平均精度の性質. **情報処理学会論文誌：データベース**, 43(SIG2(TOD13)):11–26, 2002.

[9] E. Voorhees. Variations in relevance judgements and the measurement of retrieval effectiveness. *Information Processing and Management*, 36:697–716, 2000.

[10] D. K. Harman, editor. *Overview of the Third Text REtrieval Conference (TREC-3)*, Gaithersburg, MD, 1995. National Institute of Standards and Technology.

[11] 徳永健伸. **情報検索と言語処理**. 東京大学出版会, 1999.

[12] 北研二, 津田和彦, and 獅々堀正幹. **情報検索アルゴリズム**. 共立出版, 2002.

[13] C. Buckley, J. Allan, and G. Salton. Automatic routing and ad-hoc retrieval using smart: Trec2. In D. K. Harman, editor, *The Second Text Retrieval Conference (TREC-2)*, pages 45–55, Gaithersburg, MD, 1994. National Institute of Standards and Technology.

[14] S. E. Robertson and K. Sparck Jones. On relevance probabilistic indexing and infor-mation retrieval. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[15] S. E. Robertson, C. J. van Rijsbergen, and M. F. Porter. Probabilistic models of indexing and searching. In R. N. Oddy, editor, *Information Retrieval Research*, pages 35–56. Butterworth, 1981.

[16] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In D. K. Harman, editor,

[19] http://research.nii.ac.jp/ntcir/list-resorce-ja.html

Overview of the Third Text REtrieval Conference (TREC-3), pages 109–126. Gaithersburg, MD, 1995.

[17] James P. Callan, W. Bruce Croft, and John Broglio. TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, 31(3):327–343, 1995.

[18] W. Cooper, A. Chen, and F. Gey. Full text retrieval based on probabilistic equations with coefficients fitted by logistic regression. In D. K. Harman, editor, *The Second Text Retrieval Conference (TREC-2)*, pages 57–66, Gaithersburg, MD, 1994. National Institute of Standards and Technology.

[19] Christopher D. Manning and Hinrich Schutze. *Foundations of statistical natural langauge processing*. MIT Press, 1999.

[20] J. Ponte and W. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st I Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, 1998.

[21] Scott Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[22] Susan T. Dumais. Latent semantic indexing(LSI): TREC-3 report. In D. K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 219–230. Gaithersburg, MD, 1995.

[23] J. J. Rocchio Jr. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 69–80. Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[24] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[25] Chris Buckley, Gerard Salton, James Allan, and Amit Singhal. Automatic query expansion using SMART: TREC3. In D. K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 69–80. Gaithersburg, MD, 1995.

[26] Aitao Chen and Fredric C. Gey. Experiments on cross-language and patent retrieval at NTCIR-3. In Kazuaki Kishida and Emi Ishida, editors, *Working Notes of Third NTCIR Workshop Meeting Part II: Cross Lingual Information Retrieval Task (CLIR)*. National Institute of Informatics, Tokyo, 2002.

[27] Masaki Murata, Masao Utiyama, Qing Ma, Hiromi Ozaku, and Hitoshi Isahara. CRL at NTCIR-2. In Noriko Kando, Kenro Aihara, Koji Eguchi, and Hiroyuki Kato, editors, *Proceedings of the Second NTCIR Workshop Meeting on Evaluation of Chinese and Japanese Text Retrieval and Text Summarization*, pages 5–21–31. National Institute of Informatics, Tokyo, 2001.

[28] Tetsuya Sakai, Stephen E. Robertson, and Stephen Walker. Flexible pseudo-relevance feedback for NTCIR-2. In Noriko Kando, Kenro Aihara, Koji Eguchi, and Hiroyuki Kato, editors, *Proceedings of the Second NTCIR Workshop Meeting on Evaluation of Chinese and Japanese Text Retrieval and Text Summarization*, pages 5–59–66. National Institute of Informatics, Tokyo, 2001.

[29] Yasushi Ogawa and Hiroko Mano. RICHO at NTCIR-2. In Noriko Kando, Kenro Aihara, Koji Eguchi, and Hiroyuki Kato, editors, *Proceedings of the Second NTCIR Workshop Meeting on Evaluation of Chinese and Japanese Text Retrieval and Text Summariza-*

tion, pages 5–121–123. National Institute of Informatics, Tokyo, 2001.

[30] Helen J. Peat and Peter Willett. The limitations of term co-occurrence data for query expansion in document retrieval system. *Journal of the American Society for Information Science*, 42(5):378–383, 1991.

[31] Richardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[32] Cristopher Fox. Lexical analysis and stoplists. In William B. Frakes and Ricardo Baeza-Yates, editors, *Information Retrieval: Data Structure and Algorithms*, pages 102–130. PTR Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

[33] W. B. Frakes. Stemming algorithms. In William B. Frakes and Ricardo Baeza-Yates, editors, *Information Retrieval: Data Structure and Algorithms*, pages 131–160. PTR Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

[34] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[35] 松本裕治. 形態素解析システム「茶筌」. **情報処理**, 41(11):1208–1214, 2000.

[36] R. Sedgewick. *Algorithms in C++: Parts 1-4*. Addison-Wesley, 3rd edition, 1998.

[37] 山本毅雄, 橋詰宏達, 神門典子, and 清水美都子. **全文検索：技術と応用**. 丸善, 1998.

[38] A. Fujii and T. Ishikawa. Applying machine translation to two-stage cross-language information retrieval. In *Proceeding of 4th Conference of the Association for Machine Translation in the Americas*, pages 13–24, 2000.

[39] L. Ballesteros and W. B. Croft. Resolving ambiguity for cross-language retrieval. In *Proceedings of the 21st I Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 64–71, 1998.

[40] 前田亮, 吉川正俊, and 植村俊亮. 言語横断情報検索における web 文書群による訳語曖昧性解消. **情報処理学会論文誌：データベース**, 41(SIG 6 (TOD 7)):12–21, 2000.

[41] D. W. Oard and A. Diekema. Cross-language information retrieval. *Annual Review of Information Science and Technology*, 33:223–256, 1998.

[42] 神門典子. 多言語情報の検索と利用. **人文学と情報処理**, 28:44–53, 2000.

[43] 藤井敦 and 石川徹也. 技術文書を対象とした言語横断検索のための複合語翻訳. **情報処理学会論文誌**, 41(4):1038–1045, 2000.

[44] L. Ballesteros. Cross-language retrieval via transitive translation. In W.B.Croft, editor, *Advances in Information Retrieval*, pages 204–234. Kluwer Academic Publishers, 2000.

[45] Carol Peters and Páraic Sheridan. Multilingual information access. In Fabio Crestani Maristella Agosti and Gabriella Pasi, editors, *Lectures on Information Retrieval*, Lecture Notes in Computer Science Vol.1980, pages 51–80. Springer, 2001.

[46] Michael L. Littman, Susan T. Dumais, and Thomas K. Landauer. Automatic cross-language information retrieval using latent semantic indexing. In Gregory Grefenstette, editor, *Cross-Language Information Retrieval*, pages 51–62. Kluwer Academic Publishers, 1998.

[47] D. W. Oard. Alternative approaches for cross-language text retrieval. In *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, pages 24–26, 1997.

[48] I. D. Melamed. *Empirical Methods for Exploiting Parallel Texts*. MIT Press, 2001.

[49] Jamie Callan. Distributed information retrieval. In W.B.Croft, editor, *Advances in Information Retrieval*, pages 127–150. Kluwer Academic Publishers, 2000.

[50] 岩山真, 藤井敦, 高野明彦, and 神門典子. 特許コーパスを用いた検索タスクの提案. **情報処理学会情報学基礎研究会**, (2001-FI-63):49–56, 2001.

[51] 江口浩二 and 大山敬三. 評価ワークショップによるテキスト処理研究: 第 3 回 NTCIR ワークショップを例として, 第 5 章 web 検索タスク. **人工知能学会誌**, 17(4):312–319, 2002.

[52] K. Eguchi, K. Oyama, E. Ishida, K. Kuriyama, and N. Kando. Design of Web Retrieval Task in the Third NTCIR Workshop. In *The 11th International World Wide Web Conference (WWW2002)*, number Poster-22, 1992.

[53] K. Eguchi, K. Oyama, E. Ishida, K. Kuriyama, and N. Kando. The Web Retrieval Task and its evaluation in the Third NTCIR Workshop. In *Proceedings of the 25th Annual International ACM SIGIR Conference*, pages 375–376, 1992.

[54] E. Voorhees and D. K. Harman. Overview of the sixth Text REtrieval Conference (TREC-6). In E. Voorhees and D. K. Harman, editors, *Proceedings of the 6th Text REtrieval Conference (TREC-6)*, pages 1–27. 1997.

[55] D. Hawking, E. Voorhees, N. Craswell, and P. Bailey. Overview of the TREC-8 Web Track. In E. Voorhees and D. K. Harman, editors, *Proceedings of the 8th Text REtrieval Conference (TREC-8)*, pages 131–149. 1999.

[56] N. Kando, K. Kuriyama, T. Nozue, K. Eguchi, H. Kato, and S. Hidaka. Overview of IR tasks at the first NTCIR Workshop. In Noriko Kando, Kenro Aihara, Koji Eguchi, and Hiroyuki Kato, editors, *Proceedings of the 1st NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, pages 11–22. National Institute of Informatics, Tokyo, 1999.

[57] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference*, pages 41–48, Athens, Greece, 2000.

[58] E. Voorhees. The TREC-8 Question Answering Track report. In E. Voorhees and D. K. Harman, editors, *Proceedings of the 8th Text REtrieval Conference (TREC-8)*, pages 77–82. 1999.

[59] 奥村学. 自然言語処理関連ツールあれこれ:使えるフリーソフト. **情報処理**, 41(11):1203–1207, 2000.