索引検索と同じ基準でタギングするタガーの自動生成
Perfect Synchronization of Tagging and Index Search

宅間大介[*1]
Daisuke Takuma

[*1] 日本アイ・ビー・エム株式会社 東京基礎研究所
IBM Japan Tokyo Research Laboratory

Index search and tagging are frequently used as complementary functions. When users create tagging rules, they may want to dynamically simulate those rules on the large text data using index search. Meanwhile, when using search engine, users may want to see the retrieved text with query terms highlighted by hit part tags. This paper proposes a new framework that automatically generates taggers that can use arbitrary queries of arbitrary search algorithms implemented on the framework. This new framework guarantees the consistency of the criteria of taggers and search engine without implementing taggers for each search algorithm such as word n-gram search, char n-gram search, or syntactic tree pattern matching.

1. 索引検索とタギングの整合性問題

1.1 索引検索とタギング

索引を用いた検索機能とテキストタギング機能は，実用においては，両者のペアとして必要とされることが多い．例として，文書分類用のタギングのルールを作成するケースを考えると，編集中のルールで，どんな該当文書をカバーできるか，どんな非該当文書が誤分類されるか，等を動的に調べるために，大規模な文書データ上でタギングと同じ基準で検索を行う機能が必要になってくる．一方，検索エンジンにおいても，検索文書のヒット箇所をハイライトするためにタギングが行われている．

検索エンジンとタガーは，技術的には全く性質が異なることから，通常個別に実装されており，それらの整合性を維持するのは容易ではない．実際，検索エンジンのハイライトにおいても，検索にヒットする基準とハイライトされる基準の整合性は保障されない仕様が一般的である．しかし，機密情報マスキングや保険業務における病名のタギングのように，高い recall が求められる用途では，完全に整合性が保障される検索機能でのシミュレーションが望まれる．そのため，高度な情報抽出技術を持っていても，それが検索とタギングのペアとして提供できないために，本来の効果を発揮できていないという実情がある．

以上の状況から，完全に整合性のある検索とタギングを提供することが重大な課題であると考える．

1.2 一般化されたレベルでの課題の解決

前述の課題に対する本研究のアプローチは，ある特別なフレームワークを提供し，そのフレームワーク上で作られる任意の検索アルゴリズムに対し，同じ基準でタギングするタガーを自動生成するというものである．提案の技術的な貢献は，「検索と整合性のあるタギング実装のタスクを，単語検索，文字 n-gram 検索といった個々の検索アルゴリズムではなく，転置索引型の索引検索という一般化されたレベルで解決したこと」である．

連絡先：宅間大介，日本アイ・ビー・エム株式会社，〒242-8502 神奈川県大和市下鶴間 1623-14，046-215-4415，ta9ma@jp.ibm.com

2. 索引構造の一般化と問題の定式化

2.1 転置索引とそのバリエーション

提案手法では，多くの検索エンジンで使われている転置索引の構造に着目する．転置索引は，検索対象から文書 ID への参照を持つ索引構造である．図 1に例を示す．定型項目検索索引は，文書と紐付いている項目値から文書を検索する索引である．文字 2-gram 索引は，文字 2-gram の出現文書と出現位置(offset)を保持し，2 文字以上の文字列の検索を実現する．図の例では”apple”という単語が doc 5 の offset 7-12 に登場している．単語列検索索引は，単語の出現文書と出現順序(order)を保持することで，”personal computer” (図中の doc 8)などの単語列の検索を可能にしている．より一般的には，単語だけでなく，「人名」，「製品名」等，単語のカテゴリに対して索引を持つことも可能である．構文木マッチ検索索引は，構文解析されたテキストの各単語について，構文木上の preorder, postorder, level を保持することで，「”buy”の目的語が”TV”」(図中の doc 3)といった係り受け構造の検索を可能にする．

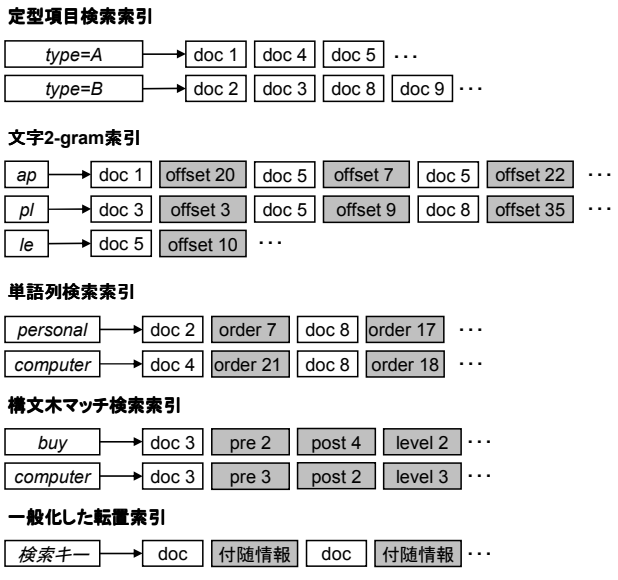

図 1 転置索引の論理構造

これらの転置索引に共通した構造を以下で参照する.

- 検索キー:索引参照キーとなる値. 図中の斜体部分.
- 付随情報:索引参照の参照先の値のうち, 文書 ID 以外の情報. 図中の灰色部分.

また, 検索キーから文書 ID と付随情報を取得する処理を「参照解決」と呼ぶことにする.

2.2 整合性問題

本研究の目標は, 前述の転置索引のみを索引として持つ検索アルゴリズムに対し, 検索機能と同じ基準でのタギングを実現することである. ここではその意味を厳密に定義する.

集合 S の有限個(0 個も含む)の元から成る集合の全体を $S[]$ と現すことにする. $S[]$の元は S の有限部分集合なので, 論理上は順序を区別しないが, 実装の観点からは S の元の配列と 捉えて理解しても良い.

検索／タギングアルゴリズムは, 以下の(K, X, f, Q, g)の 5 つ組と定義する. ここで D は, 文書全体の集合とする.

1. 検索キーの全体の集合 K: 前述の転置索引の検索キーの全体で, 任意の集合を想定する.
2. 付随情報の全体の集合 X: 前述の転置索引の付随情報の全体で, 任意の集合を想定する.
3. 前処理写像 f: $D \rightarrow (K \times X \times P)[]$: 文書から(検索キー, 付随情報, 抽出元オフセット情報)のトリプレットを 0 個以上抽出する処理に相当する. ここで, P はオフセットを表す集合 $\{(i,j) \mid i,j$: 整数, $0 \leqq i < j\}$である.
4. クエリの集合 Q: 検索／タギングアルゴリズムが受理するクエリの全体を表す.
5. 情報抽出写像 g: $Q \times D \rightarrow ((K \times X \times P)[])[]$: $g(q, d) \in f(d)[]$を満たすものとする. クエリ q と文書 d が与えられた時, その文書から抽出された(検索キー, 付随情報, 抽出元オフセット情報)のトリプレットの集合 $f(d)[]$から, クエリにマッチする組み合わせを 0 個以上列挙する処理に対応する.

以下に, 例として, 単語列検索のケースにおける検索／タギングアルゴリズム(K, X, f, Q, g)の定義を記す. ここで挙げる単語列検索は一語のワイルドカード「*」も解釈可能なもので, 図 1 の単語列検索索引によって実現可能なものとしている.

1. 検索キーの全体の集合 K: 単語分割処理が出力する単語の全体(未知語も含む). 全ての文字列としてもよい.
2. 付随情報の全体の集合 X: 1 以上の整数{1,2,3,…}とする. 文書中の単語の出現順序を表す.
3. 前処理写像 f: $D \rightarrow (K \times X \times P)[]$: 入力文書 $d \in D$ に対し単語分割結果を対応させる. 例として, d =「彼は赤い服を着る。彼は青い服も着る。」は $f(d)$ = { (「彼」, 1, 0-1), (「は」, 2, 1-2), (「赤い」, 3, 2-4), …, (「。」, 14, 17-18)} に写像される. ここで, オフセット $(i, j) \in Y$ は i-j と記している.
4. クエリの集合 Q: $(K \cup \{「*」\})$の元の長さ 1 以上の順列の全体とする. 例として長さ 3 の順列「服」,「*」,「着る」は Q の元である.
5. 情報抽出写像 g: $Q \times D \rightarrow ((K \times X \times P)[])[]$: $(q, d) \in (Q \times D)$とする. q の「*」以外の要素とその出現順序の組を $(k_1, y_1), (k_2, y_2), \cdots, (k_n, y_n)$で表す時, $f(d)$の部分集合 $\{(k_1, x_1, p_1), (k_2, x_2, p_2), \cdots, (k_n, x_n, p_n)\}$で, $(\exists z$:整数$)(y_1=x_1+z, y_2=x_2+z, \cdots, y_n=x_n+z)$ となるもの全てを返す. 例として d=「彼は赤い服を着る。彼は青い服も着る。」, q=「服」,「*」,「着る」の場合, $(k_1, y_1), (k_2, y_2), \cdots, (k_n, y_n)$ は (「服」, 1), (「着る」, 3) となり, $g(q, d)$ ={{(「服」, 4, 4-5), (「着る」, 6, 6-8)}, {(「服」, 11, 13-14), (「着る」, 13, 15-17)}} となる.

このように定義した検索／タギングアルゴリズムの定義によって, 検索, タギングはそれぞれ以下のように表現できる. 検索は, ID 付き文書集合{(1, d_1), (2, d_2), …, (n, d_n)} (d_1, d_2, …, $d_n \in$D, 1, 2, …は ID) があり, クエリ $q \in Q$ が与えられた時, $\{i \mid i$ は 1 以上 n 以下の整数, $g(q, d_i)$ の元の数が 1 以上$\}$をヒット文書 ID 集合として返す処理を検索と定義する. タギングは, 文書 d とクエリ q(タギングにおいては「ルール」に相当する)が与えられた時, $g(d, q)$ の各元 $\{(k_1, x_1, p_1), (k_2, x_2, p_2), \cdots, (k_n, x_n, p_n)\}$ について, d のテキスト上の $p_1, p_2, \cdots p_n$ の最小開始オフセットから最大終了オフセットまでの範囲に q でラベルづけされたタグをつける. 厳密には複数のタグがオーバーラップする可能性があるため, 機能としては, アノテーション付与と捉えても良い.

これらの表現によって, 検索とタギングが同一の検索／タギングアルゴリズムに基づいて定義されることを, 「整合性がある」と呼ぶことにする. 上記の定義は暗に検索／タギングの機能範囲を制限しているが, この表現可能範囲については3.3 提案フレームワーク上で実現可能な機能のセクションで触れる.

3. フレームワーク

3.1 データフローと中間形式

本研究で提案するフレームワークは, 索引検索, タギングの機能を実装する際に, データが経由すべき中間形態を定義したものである. オブジェクト指向言語で言うと, 索引生成, 検索, タギングの処理のインターフェースを定義することに相当する. 整合性問題の解決は, 検索／タギングアルゴリズムに相当する部分が, 同一の処理に集約されていることと対応する.

前述の「*」によるワイルドカード機能付きの単語列検索を例に, 図 2を用いてデータフローの概要を述べる. ポイントとなるのは, 中間情報(A), 中間情報(B)と呼んでいる中間状態を出力する処理を課していることである.

索引生成では, 逐次入力される文書から検索キーと付随情報を抽出し, 検索キーから(文書 ID, 付随情報)への参照を転置索引として生成する. この処理において, 提案フレームワークでは, 中間情報(A)として索引に保持する全ての付随情報についての(検索キー, 付随情報, オフセット情報)のトリプレットを返す実装を要請する. これは検索／タギングアルゴリズムの定義における前処理写像 f そのものに相当する.

検索／タギングは, 情報抽出写像 g で定義される処理を行う. 整合性問題の定義から, 入力のクエリは共通となる. この処理において, 提案手法では, クエリの情報を分解した中間情報(B)を生成することを要請する.

中間情報(B)は, クエリの処理において転置索引での参照解決が必要な検索キー(B1)と, 参照解決結果から検索結果／タギング結果を生成するのに必要な情報である検索キー制約(B2)から成る. 単語列検索の例における「服 * 着る」というクエリからは, 「服」, 「着る」が検索キー(B1)として抽出され, 検索キー制約(B2)は, アスタリスクを 1 単語分とカウントして, 「着る」の出現順序が「服」のそれよりも 2 大きい, という制約になる. 検索キー(B1)は, 検索時は転置索引により参照解決され, タギング時は中間情報(A)を 1 文書の転置索引とみたてて参照解決される.

検索キー制約の適用は, 検索／タギング共通で, 検索キー制約(B2)と参照解決結果を入力とし, 情報抽出写像 g の写像先に相当する (検索キー, 付随情報, オフセット情報)のトリプレットの集合の集合を出力する. タギング時は入力の参照解決結果の(検索キー,付随情報)の 1 つ 1 つにオフセット情報が紐付

いているため，検索キー制約を満たす(検索キー,付随情報)の組を算出した上で，対応するオフセットと組み合わせたものを出力する．検索時は，参照解決結果を文書 ID ごとに分離した上で，同様の検索キー制約を適用して (検索キー,付随情報)の組を算出する．オフセットは転置索引上に存在せず，検索結果上でも使われないため，入力，出力ともにダミーの値でよい．ここで，検索キー制約適用処理が，検索時も参照解決結果全体に対する処理ではなく，個別の文書 ID ごとの処理に分解可能であることは，本質的に重要である．これは，g が $Q\times D[]$ではなく $Q\times D$ 上の写像として well-defined であり，各文書の検索の該当／非該当が他の文書に影響されないことと対応する．この条件を満たさない検索方式については3.3 提案フレームワーク上で実現可能な機能のセクションで触れる．

検索キー制約適用処理の出力は，タギング時はそのままタグ情報となり，検索時は，文書 ID のユニークを取ることで検索結果となる．

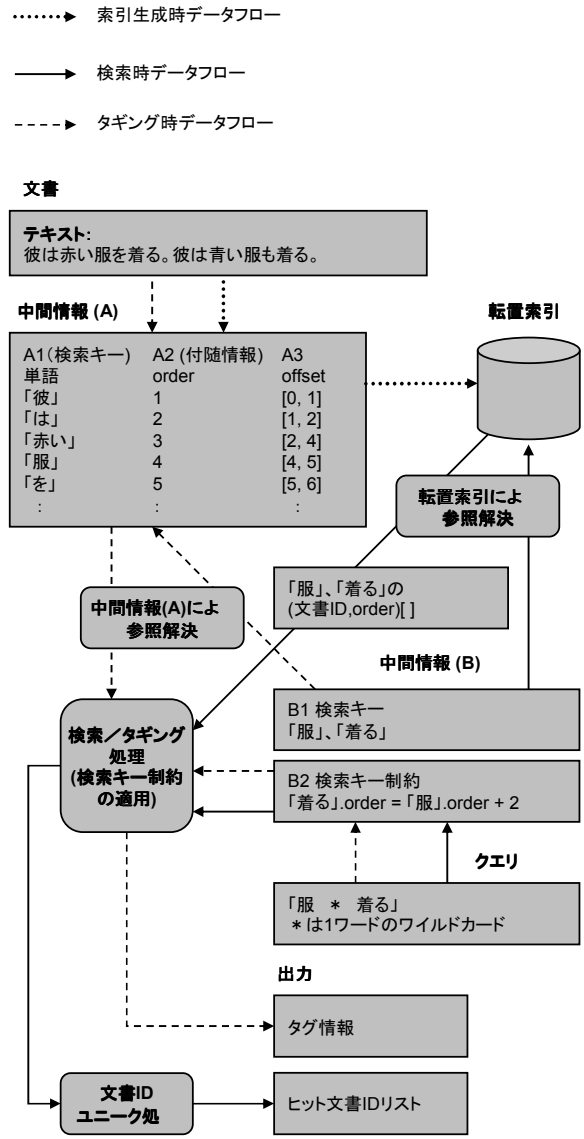

図 2 データフロー

具体的なインターフェースとしては，以下の定義が考えられる．1 つの検索／タギングアルゴリズムについて以下を実装することで整合性のある検索／タギング機能を提供できる．

1. 中間情報(A)の生成： 文書を入力とし，中間情報(A)を出力する．
2. 索引生成： 中間情報(A)のストリームを入力とし転置索引を生成する．
3. クエリ分解： クエリを入力とし，中間情報(B)を生成する．
4. 参照解決： 検索キーを入力として，参照解決結果を返す．このインターフェースのみ中間情報(A)を用いる実装と転置索引を用いる実装の 2 つの実装が存在するが，これらの実装は検索／タギングアルゴリズムによらないためフレームワークレベルで共有できる．フレームワークが提供する機能と考えても良い．
5. 検索キー制約適用： 1 文書分の参照解決結果と検索キー制約(B2)を入力として，(検索キー，付随情報，オフセット情報)のトリプレットの集合の集合を出力する．

3.2 整合性の保障

提案フレームワーク上では，検索とタギングが同じ情報抽出写像 g によって定義されることが保障されている．実際に任意の文書 d と任意のクエリ q が与えられたときを考える．d から生成される中間情報(A)，q から生成される(B)は検索とタギングで共通になる．転置索引は d 以外の文書を含むが，任意の検索キーk について，文書 d を含む参照解決結果($d.id$, d の中の k の付随情報，ダミーのオフセット)[] は，d の中間情報(A)を用いて k を参照解決した結果と，オフセット情報の有無を除いては一致する．検索キー制約適用の処理は，d 以外の文書の影響を受けず，入力となる参照解決結果と q のみで一意に決まるように定義されており，g の写像先は，検索時，タギング時でオフセットを除いて等しくなる．

また，実装面で g の定義に関する部分で検索とタギングで別個のロジックが実装され得ないことも確認する必要がある．これについては，唯一検索時とタギング時で別の実装が呼ばれる参照解決処理が，個々の検索／タギングアルゴリズムによらずフレームワークレベルで共通化できることから保障される．

3.3 提案フレームワーク上で実現可能な機能

提案手法は，提案フレームワークで実現可能な検索とタギングについてのみ，整合性を保障するものであるため，この提案フレームワークのサポート範囲について考察する必要がある．提案フレームワークは，少なくとも以下の処理で実現できる検索であれば記述可能である．

1. テキストを 0 個以上の(検索キー，付随情報)に変換する．
2. 所与のクエリに含まれる検索キーを参照解決する．
3. 参照解決結果の(文書 ID，付随情報)に対し，個別の文書 ID ごとの情報のみから，当該文書がヒット判定，および，ヒットに関与した検索キーの特定を行う．

具体的には， 図 1の転置索引で実現される定型項目検索，文字 n-gram による全文検索(文字列の検索)，単語列検索，部分構文木検索が実現可能である．逆に実現できないケースとしては，上記の 3 番目の処理に相当する箇所で，文書のヒット基準がその文書自体の情報だけで決まらないタイプの検索が挙げられる．例としては，検索キーについて TF や IDF を用いたスコアを算出し，スコアが閾値を超えた場合に文書をヒットさせる検索がこれに該当する．これは，TF や IDF が個別の文書だけではなく，検索対象の文書全体における検索キーの出現数に依存するためである．このような検索は，文書 $d\in D$ の該当／非該当が他の文書にもよってしまうため，情報抽出写像 g が $Q\times D$ 上の写像としては well-defined でなく，整合性のあるタガー自体が存在し得ない．本研究における検索／タギングアルゴリズ

ムの定式化はそのような検索を排除するように定義したことになる.

3.4 開発と機能への影響

提案フレームワーク上に検索機能を実装する場合, 前述のインターフェースの実装が必要になるが, 処理の流れは通常の検索エンジンと同様で, オーバーヘッドは中間情報の入出力のみであるため, 事実上開発コストの差分は少ないと考える. またパフォーマンスについても, ボトルネックである参照解決部分は, 転置検索を用いた一般的な検索エンジンと変わらないため検索側は影響がない.

4. 関連研究

本研究で意図しているタギングに関連した分野としては, 古くから研究されているフレーズレベルの属性を用いた文書分類[Lewis 1992] [Furnkranz 1998]や, 固有表現抽出, イベント抽出 [Riloff 1993] [Huffman 1996]が挙げられる. これらの分野に共通して言えることは, 情報抽出のルールは自動生成されるが, 運用の段階で更なる精度向上が必要になった場合に人手でのルールのメンテナンスが重要になるということである. 実際に文書分類においては, 抽出精度に加えて, ルールの可読性／編集可能性も手法の優位性に挙げている例[Antonie 2002]がある. 本研究は, 特定の情報抽出ルール生成手法に対し, 生成され得るあらゆるケースのルールでの検索を可能にしておくことで, メンテナンス段階において生成されたルールをダイナミックにチューニングすることを保障する.

一方, 検索エンジンにおける検索結果の可視化については, web 検索の分野で可視化技術の分類がなされており[Mann 1999], 自己組織化マップなどの文書集合レベルの可視化, web サイトレベルの可視化, ヒット文書レベルの可視化に分類した時, ヒット箇所のハイライトに相当するヒット文書レベルのエリアは, [Hearst 1995]等の研究はあるものの, 全体としては他のレベルに比べ研究例は少ない. 実際に実用化されている検索エンジンにおいても文書レベルの可視化としては, 検索ヒット箇所周辺の要約, ヒット箇所のハイライトが定着していて, それほどバリエーションは無い. また, 個々の検索ロジックのハイライト処理については実装のみで解決可能な問題であるため, 研究の対象とはなっていなかった.

5. 結論

5.1 整合性問題の定式化と解決

本研究では, 検索とタギングが「整合性がある」ということを厳密に定義し, 重複開発することなく検索とタギングの両方を実装するためのフレームワークを提案した. このフレームワークでの実装可能範囲は,「整合性があること」の十分条件であり, それでいて, 既存の多くの検索ロジックがカバーされている. 検索の視点からは, 転置索引で実現可能という前提の他に, 文書が検索結果に含まれるかどうかがその文書内の情報のみよってきまることが整合性確保可能性の本質的な条件であった. 一方, タガーを作成する立場では, 検索ロジックを実装することでタガーを自動生成するという都合上, 索引構造を意識する必要はでてくるが, 通常では全くサポートできないダイナミックなメンテナンス, シミュレーションという付加価値が得られる.

5.2 課題

本研究で提案した整合性の定式化やフレームワークは, 検索, タギングの結果の一貫性のみを問題意識としてきたが, 実際に検索, タギングを実現するには索引サイズや検索コストも考慮する必要がある. これらの観点が重要になる例としては, 任意の 1 文字の数字(正規表現「[0-9]」)のような特定の文字セットでのマッチをサポートする文字 *n*-gram 検索が挙げられる. クエリ拡張でこれをサポートする場合, クエリから検索キーを抽出する段階で,「[0-9]」→「0」,「1」, …,「9」のように文字セットが個別の文字に展開されることになり, 検索コストにインパクトがある. 一方, 索引上に「一文字の数字」を意味する文字を含む *n*-gram を検索キーとして追加する方法でこれをサポートすると, 数値の連続があった場合に索引サイズが増加する. こうした問題を回避するためには検索／タギングアルゴリズムのクエリに何らかの制限が必要になるため, 現実的に実現可能な機能のクラスは本研究で定義した検索／タギングアルゴリズムの機能のクラスよりも小さなクラスになる. しかし, この「実現可能な機能のクラス」の定義は, 本研究で定義した検索／タギングアルゴリズムのように一意に定義するのではなく, パフォーマンス要請を与えた上で定義すべきものなので, 本質的に別のアプローチが必要であると考える.

参考文献

[Lewis 1992] David D. Lewis: An evaluation of phrasal and clustered representations on a text categorization task, ACM SIGIR, 1992.

[Furnkranz 1998] J. Furnkranz, T. Mitchell: A case study in using linguistic phrases for text categorization on the WWW, AAAI/ICML Workshop on Learning for Text, 1998.

[Riloff 1993] E. Riloff: Automatically constructing a dictionary for information extraction tasks, AAAI, 1993.

[Huffman 1996] S. B. Huffman: Learning information extraction patterns from examples , Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 1996.

[Antonie 2002] Maria-Luiza Antonie, Osmar R. Zaïane, Text Document Categorization by Term Association, IEEE, 2002.

[Mann 1999] Thomas M. Mann, Visualization of WWW-Search Results, Database and Expert Systems Applications, 1999.

[Hearst 1995] Marti A. Hearst, TileBars: Visualization of Term Distribution Information in Full Text Information Access, ACM CHI, 1995.