社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－NL－185 (12)
2008－SLP－ 71 (12)
2008／5／23

集合知を利用した語彙情報の収集・共有・管理システム

中野 鐵兵† 佐々木 浩† 藤江 真也† 小林 哲則†
†早稲田大学

あらまし:
音声・言語アプリケーションにおける従来の語彙情報作成手法の問題点を解決するため，集合知を利用した語彙情報の収集・共有・管理システムを提案する．具体的には，語彙情報を集中管理するためのオンラインデータベースシステムを構築し，それを利用者に公開する．提案システムでは，Web資源からの語彙情報の自動収集の枠組みを備え，データの集約を図る．また，アプリケーション用語彙の新規作成から，その継続的な更新まで包括的な解法を提供し，これまで各々の開発者がアプリケーション毎に用意していた語彙定義のプロセスの一元化を図る．さらに，インタフェースを広く公開し，アプリケーション間の語彙定義の共有や，アプリケーションで使用する語彙の自動更新のサポートを図る．本稿では，実際に提案システムの実装として開発されたプロトタイプシステムと，提案システムによって実際に有効な語彙リストの生成が可能である事を示した評価実験について述べる．

Lexical Data Collection, Sharing, and Management System using Collective Intelligence

Teppei NAKANO†, Hiroshi SASAKI†, Shinya FUJIE†, Tetsunori KOBAYASHI†
†Waseda University

Abstract:
In order to solve the problems of the conventional approach of designing lexicons, we propose a new approach: using a lexical data collection, sharing, and management system using collective intelligence. In particular, we construct and operate a new online database system for lexical informations. The proposed system is designed as a data intensive system so that it can collect lexical information from all web-based resources. Also, the system provides the comprehensive solution of designing lexicons so that the designing processes of lexicons can be standardized. Besides, the system interface is published so that lexical informations are shared by many applications. In this paper, the prototype system developed based on the proposed approach and the feasibility test for designing lexicons are described. The assessment result showed that the proper lexicons can be generated from the proposed system.

1 はじめに

音声認識アプリケーション開発における最も重要かつ困難な作業の一つとして，システムが認識可能な語彙の適切な設計と，実際に利用されている語彙のメンテナンスが挙げられる．

適切な語彙はアプリケーション依存で決定されるため，開発者はそのアプリケーションに適した語彙を個別に設計する必要がある．例えば，スポーツ番組のディクテーションを行う際には，スポーツ関係の語彙を，レストランナビゲーション用の音声認識システムを構築する際には，料理関係の語彙や住所，レストラン名を網羅的に用意する必要がある．しかしながら，このような特定の属性をもった語彙を網羅的に設計する枠組みは存在せず，非効率な手作業が必要となる．また，VoiceNote[2] のように，音声コマンドによる操作が可能なアプリケーションを構築する場合，ユーザビリティの観点から標準的な音声コマンドを念頭に置いた語彙設計が求められる．しかしながら，音声コマンドの語彙の標準化は未だ進んでおらず，実際に利用される語彙を開発者が事前に推定するのも難しい [1]．過去に作成された類似タスクの語彙情報の流用しようとしてもそれらの情報の共有に関する枠組みは存在せず，過去に作成された語彙情報の活用も困難である．さらに，適切な語彙の収集や適切かどうかの判断，読み情報の付与はアプリケーション毎に個別に人手で行われ，これらを効率的に解決するための枠組みは用意されていなかった．

アプリケーション用に設計した語彙は一度設計したら完成というものではなく，実際に適切な語彙情報の利用を可能にするためには，継続的な語彙情報のメンテナンスが必要となる．例えばPodcastle[3]のような，新規性・話題性の高い単語が多く現れる音声を対象としたディクテーションシステムでは，関連する話題のコーパスをウェブから収集したテキストを用いて逐次更新し，言語モデルと語彙を頻繁に更新することで音声認識の精度向上を図っている．またクラスN-gramを用いたアプリケーションの場合，クラス毎に生起する単語を厳密に定義する必要があるが，開発時に定義した語彙やその生起確率が全く変化しないとは考えにくい．さらに，カーナビゲーションシステムのような音声認識アプリケーションでは，住所名の変更や新規の施設名への対応が求められる．このような日々増殖・変化する語彙を逐次更新するため，ウェブ上の情報資源を基に語彙リストを構築する手法がとられることがある．しかし，ウェブ上で提供されるサービスは語彙情報取得に特化しているわけではなく，そこから必要な情

報の抽出が必要である．さらにこれらの枠組みは，アプリケーション開発毎に個別に設計・実装される必要があり，作業負荷や精度・効率の面で問題がある．また，更新された語彙情報を実利用環境に配信するための枠組みも求められる．

そこで本研究では，アプリケーションに用いる語彙情報作成の負荷を低減するため，語彙情報をアプリケーション開発者で共有する枠組みを提案する．そのためにあらゆる分野の語彙情報を一元化されたオンラインデータベース上に蓄積し，共有のウェブサービスとして提供する．音声認識アプリケーションの語彙情報の管理にウェブシステムを利用した例として，w3voice[4] や MusicNavi[5] がある．w3voice では音声認識アプリケーションを Web サービスとして共有し，ユーザの作成した Web サイトから利用できるようにする枠組みを提案している．また，音声認識用辞書を共有することにより，認識用語彙情報のユーザ間での更新を実現している．MusicNavi では楽曲に関する語彙の辞書をオンラインデータベースの形で共有し，音声認識システムに用いている．しかし，これらのシステムで蓄積された語彙情報は指定されたアプリケーションのみでしか扱えず，ユーザの作成したアプリケーションに組み込むための枠組みは提供されていない．また，音声認識・言語処理アプリケーション全般を対象とした語彙情報の共同管理の枠組みは存在しない．それに対して本研究では，自由に利用可能な形式で語彙情報を集約し，簡単な要求を投げるだけで必要な語彙情報を得ることができるようなウェブベースの枠組みを実現する．また，クローラによる Web 資源からの自動収集の枠組みと，利用者の集合知を利用した半自動的な語彙情報作成の枠組みを構築し，データベースの増強を図る．さらに，利用者の要求を保持し，データベース上の語彙の情報が更新されたり，新規の語彙が追加された際に利用者やアプリケーションへ反映される機構を用意する．この枠組みを Proxy-Agent[6] のような音声認識システム拡張の枠組みと組み合わせることで，語彙情報の動的な更新が可能な音声認識アプリケーション開発の新しい枠組みの実現を目指す．

本稿では，次節でまず提案システムの基本的なアプローチについて述べる．次に提案システムを実装する上で発生する課題と，その対応方法について述べ，4 節で提案システムによる語彙情報共有のシナリオを説明する．5 節で，提案システムの実装として開発したプロトタイプシステムについて述べ，6 節で，提案システムにおいて，実際に必要な語彙情報の取得が可能かを検証した実験と，その結果にについて報告する．

2 基本アプローチ

本研究では，これまで各々の開発者がアプリケーション毎に用意していた語彙定義のプロセスを一元化する枠組みを構築し，語彙情報作成の効率化と作成された語彙の高精度化，またそれに伴う知見の集約を図る．具体的には，語彙情報を集中管理するためのオンラインデータベースシステムを構築し，それを利用者に公開する．分野や品詞，粒度 (語を構成する単位の大きさ：単語や複合語，助詞を含んだ節) にとらわれない，多種多様な語を対象とする．語彙情報としては，語とその読み情報，その語のメタ情報を扱えるようにする．また，利用者による語彙情報の追加・修正を可能にし，集合知を利用した語彙メンテナンスの実現を図る．以下に提案システムの基本的なアプローチを述べる．

Data Intensive Systems 音声・言語アプリケーションで必要な語彙情報を提案システムに集約し，その利用価値を高める．そのために，単一の語の読み情報の取得から，アプリケーション用語彙の作成・管理まで，語彙情報に関連する全ての作業を提案システムで完結できるようにする．またシステムを広く公開し，自然と語彙に関連する情報が集約されるような枠組みを提供する．

Lexicon Lifecycle アプリケーション用の語彙の新規作成から，その継続的な更新まで包括的な解法を提供する．そのために，システムが保持する膨大な情報元から利用者が必要とする語を選択する枠組みを構築する．また，新しくデータベースに追加された語彙 (新着語と呼ぶ) から，必要とする語彙のみを効率的に扱える枠組みをあわせて構築する．

Cooperative Framework 語彙情報を必要とするアプリケーション同士のゆるやかな連携を可能にする．すなわち，アプリケーションで使用する語彙の定義と追加・修正された語彙情報の共有を可能にする．また，インタフェースを広く公開し，音声・言語アプリケーションに限らず，さまざまなアプリケーションからの利用を可能にする．これにより，他の多くアプリケーションで利用される語彙情報や語彙定義情報へのアクセスを実現し，自然発生的な標準語彙の利用を可能にする．

以降，これらの特徴を満たしたシステムを構築するための，本研究における具体的な実装のアプローチを述べる．

2.1 WWW 上の語彙資源の利用

例えば利用者が標準語彙やレストラン名の一覧，地名の一覧など様々な語彙を取得できるようにするため，データベースは初期の段階で十分な量の基本語彙とその語彙に含める情報が用意されている必要がある．標準語彙の整備には例えば ipadic[1] などの WWW 上で入手可能な既存の辞書を活用する．また，ユーザのアップロードによる語彙追加の枠組みも設け，語彙の増強を図る．加えて，レストラン名の一覧などの日々更新される語彙については，システムが情報源を随時巡回する機能を持つ必要がある．それを実現するため，クローラを用いて WWW 上の語彙資源から随時語彙情報を収集するようにする．こうすることにより，データベース内の語彙の新規性の維持や開発者間での標準語彙の共有が実現できる．他所を巡回し，その情報を引用する際には語彙

[1] ipadic version 2.7.0,http://chasen.naist.jp/stable/ipadic/ipadic-2.7.0.tar.gz

情報の収集元の情報も保持，明記し，権利上の問題に配慮する．

2.2 クエリベースの目的語彙の選別

例えばシステムがデータベースに蓄積した様々な語彙から，施設名の一覧のみを取得するなど，特定の語彙の集合のみを取り出せる必要がある．データベース内の全ての語を利用者が1つずつ判定して取り出すようなことは事実上不可能であるため，特定の語彙の集合を端的に表現する手段を用いて目的語彙を選別しなくてはならない．また，新着語についても同様で，新着語の中で目的語彙に沿った語のみが利用者の持つ語彙に追加される必要がある．例えば施設名の一覧には施設名のみが新しく追加されることが好ましい．そういった観点から，データベースから必要な語彙を選別するための手段そのものを特定語彙を表す手段として用いて，全ての新着語から必要な語のみを獲得する枠組を定める．

大量の情報から目的の情報を得る方法でまず考えられるのが，既存の多くの検索エンジンで採用されている，検索条件としてのクエリを用いたものである．検索条件の結果を語の集合とすれば，語の集合をクエリで表しているといえる．このような機構を実現するためには，検索の際に用いられるメタ情報を語彙自身が持つ必要がある．Yahoo!カテゴリ[2]はWebページへのリンクにクラスの情報を持たせ，そのクラス情報を基にWebページへの参照を提供するサービスである．これを語彙取得に応用し，“クラスタリングにより語彙にクラスの情報を持たせ，検索の際にクラスを指定し，そのクラスに属するような語彙のリストを作る．”といった形が考えられるが，不特定多数のアプリケーションに対応するため語彙の一意なクラスタリングは難しい．そこで一意なクラスタリングを用いず，語彙に対してのメタ情報として自由なタグ付けを許し，そのタグ名とクエリの一致によって語彙を選別する方法をとる．その際のタグの情報は既存の語彙辞書やユーザ定義，Webなどの分類の情報から幅広く回収し，多くのクエリへの対応を図る．

2.3 ウェブベースのアプリケーション連携

例えばある利用者がホテル名の一覧を作成するとき，他の利用者が一度ホテル名の一覧を作成していたならば，その情報を有効に活用できる枠組みが求められる．すなわち，他のアプリケーションに用意した語彙と語彙定義の情報を他のアプリケーション用の語彙設計時に利用できる機能を備え，その機能を用いて効率的な語彙設計をサポートするシステムが必要である．提案システムでは，利用可能性を高めるために，ウェブベースのアプリケーションとしてこの語彙設計用アプリケーションを構築する．これらを実現するため，データベースには利用者が語彙リストを作成したときに用いたクエリと追加・削除した語彙の情報を併せて保持する．他の利用者が同じクエリによって語彙の検索を行ったときにはこれらの情報を提示し，直接利用できるようにする．また，語彙リスト生成時に保持した情報を用いて，新しくデータベースに追加された語を選別し，利用者に通知する．利用者は通知された語を語彙リストに追加ができる．追加された際，他の利用者がその語彙リストを利用していた場合は，その利用者に対して語彙リストに追加された語を新しい語として通知する．このことにより，語彙情報作成の負荷を低減させるだけではなく，新着語の追加などの語彙の管理を分散させることができる．

語の読みの情報などの語彙に含まれる情報に不備があった場合に備え，利用者が効率的に不備を修正し，その結果を共有できる枠組みを用意し，データベースの正確性の向上させる必要もある．これを実現するために，利用者によって修正された語も新着語同様に他の利用者に通知する機能を持たせる．

さらに，すでに運用中のアプリケーションに対して，更新された語彙の配信を可能にする枠組みも必要である．すなわち，アプリケーションを対象としたインタフェースを備え，提案システムと音声認識アプリケーション間の連携を可能にする．提案システムでは，アプリケーション同士のゆるやかな連携を可能にするため，このインタフェースをウェブサービスとして提供する．特に，Proxy-Agent[6]のプラグインとしてこのウェブサービスとの連携を実装することで，音声認識エンジン・アプリケーションによらない汎用的な語彙情報更新の枠組みの提供を可能にする．

3 提案システムの実現における課題とその対応手法

3.1 語の粒度不均一問題

様々な情報源から語の粒度を定めずに語彙を収集するため，比較的汎用性の低い，粗い粒度の語が多く含まれる可能性がある．例えばデータベース内に「早稲田大学理工学部」といった語があるにもかかわらず，「早稲田大学」や「理工学部」などのより汎用性の高い語が存在しないなどである．このようなことはデータベースとしての有用性を損ねてしまう．

この問題に対し，データベース上の粒度の粗い語はより小さい粒度の語に分割し，分割された語を新規語彙として利用する(ここでは細粒度語と呼ぶ)．最も細かい粒度の語の単位は形態素とする．

3.2 タグの付与指針不統一問題

様々な情報源からタグの情報を収集した場合，タグの付与指針が異なるため，検出率が低下する恐れがある．例えばタグ情報としてオンライン百科事典Wikipedia[3]の記事のカテゴリ情報を採用して，「早稲田大学」に「東京都の大学」をタグ付けし，「明治大学」には他の情報源によって「大学」をタグ付けしていた場合などである．このようなことは検出率の低下を引き起こす．例えば「大学」のタグで「早稲田大学」が検出ができない．

この問題に対し，タグの細粒度化とタグの伝搬というアプローチをとる．タグ情報となる語にはその上位概念となる語が多いが，そういった語の名詞を

[2]Yahoo!カテゴリ, http://dir.yahoo.co.jp/

[3]Wikipedia,http://ja.wikipedia.org/wiki/

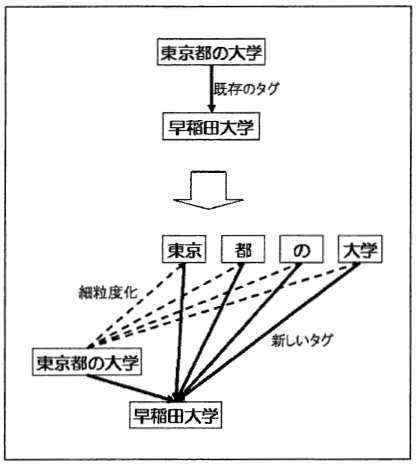

図 1: タグの細粒度化とタグの伝播

含んだ細粒度語が，より上位の概念を表すという傾向がある．例えば「東京都の大学」に対しては「東京」「大学」「東京都」「東京都の」「の大学」がより上位の概念を表す語となる．この傾向を利用して，語彙に対して付与された，タグとなる語の細粒度語も，タグ情報として利用する．例えば「早稲田大学」に「東京都の大学」がタグ付けされた場合には，その形態素「東京」「都」「の」「大学」もタグ情報として保持する (図 1)．この情報を基に「東京都の大学」の各細粒度語をタグ情報として利用できるようにする．こうすることにより，粒度の細かい語をタグとして利用することができる．

3.3 利用者クエリとタグ情報の乖離問題

利用者はデータベース内のタグの付与状態を把握していない上，タグの情報はクローラによって随時更新されるため，作成した目的語彙に対して入力したクエリが最も適切にその語彙を表しているとは限らない．つまり，その後の新着語の獲得が適切になされないということになる．例えば，東京都の大学名の一覧を最も適切に表すクエリが「東京 and 大学 and 施設」であったとする．この時，利用者が東京都の大学名の一覧を取得するために「東京 and 大学」をクエリとして指定した場合，その後の新着語獲得において「and 施設」で取り除かれるべき語も取得してしまう．

これに対し，システムによるクエリ再生成の枠組みを導入するアプローチをとる．この枠組みでは，利用者が編集した語彙定義情報を表すものとして利用者が入力したクエリをそのまま用いるのではなく，編集した語彙を基にシステムが判定した，そのデータベース上で最も適したクエリを用いる．このシステムが再生成したクエリを利用者に通知することで，利用者クエリとタグ情報の乖離を最小限にする．こうすることにより，前述した例で利用者が「東京 and 大学」を入力したとしても，システムが利用者の編集状態より「東京 and 大学 and 施設」を判定するため，利用者は最も適した新着語獲得規則を利用することができる．さらに，システムがクエリの生成作業を常時行うことにより，随時更新されていくタグの情報にも対応することができる．

3.4 語彙定義基準の不透明問題

他の利用者の語彙定義情報を利用する際，その語彙選択の基準が明確でないと利用が困難になる．例えば，他の利用者が「東京 and 大学」で出力したリストを編集したものだと分かったとしても，その編集が東京都の大学名という基準で編集したものであるのか，東京都の大学に関係するものという基準で編集したものであるのか，または違う基準で編集したものであるのかは不明である．将来の新着語獲得にも影響を及ぼすため，このような語彙選択基準の曖昧なものを採用することはシステムの正確性の低下を招いてしまう．

これに対し，利用者が編集した語彙情報をそのまま共有するのではなく，その語彙を表すクエリを共有する方法をとる．これにより，語彙の選択基準が明確になり，情報の曖昧性の問題が解決される．利用者の編集した語彙とクエリで出力される語彙とに生じる差は，システムによるクエリ再生成において最小限にされる．

4 語彙情報共有のシナリオ

システムを構築するに当たっての語彙情報の整備のシナリオを以下に示す．

1. 語の読みの情報と分類の情報を既存の辞書や Web リソースから収集
2. 分類の情報をタグ情報とし，情報源の情報と共にデータベースに登録
3. 語を形態素に分割し，タグ情報として利用
4. 1〜3 を繰り返し，語彙を拡充し続ける

次に語彙リストの生成と共有のシナリオを示す．

1. クエリに指定されたタグ条件を基に語彙のリストを出力
2. 出力結果を編集し，保存する
3. ユーザの編集した結果はクエリの形で他ユーザから利用可能になる
4. 3 で編集した結果に近いクエリを生成
5. 生成したクエリを規則として，2 の結果に新規語彙を追加し，通知する
6. ユーザは新規語彙のみを編集
7. 4〜6 を繰り返す
8. 3 でリストが他ユーザから利用されていた場合はその編集が逐一反映される

5 プロトタイプシステムの開発

前節で述べたシナリオをサポートする，プロトタイプシステムの開発を行った．図 2 に概観を示す．本システムは Web アプリケーションとして動作し，データベースの利用は Web ブラウザ上で行う．

データベースでは，語の綴りの情報，読みの情報，収集元の情報を保持し，語の綴りの情報から読みや収集元の情報への関連を持つようにした．タグも語

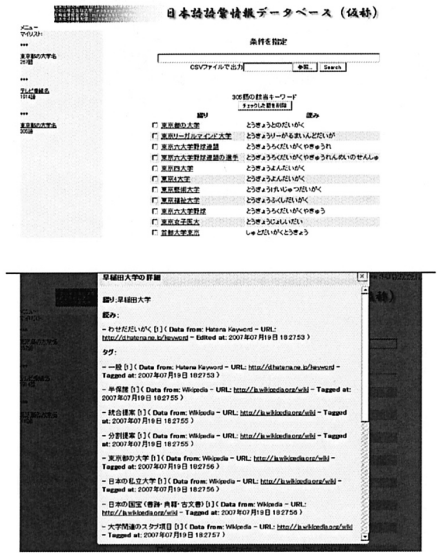

図 2: 作成したシステム．(Top) クエリの入力，不適切語の削除及びリストの名前入力画面．(Bottom) 語の詳細情報提示画面

として登録され，語と語との関連としてタグを定義するようにした．また，タグには役割を表す語を関連付けられるようにした．例えば品詞を表すタグには “品詞” の語への関連を持つようにする．

他の開発者が定義した語彙情報を共有するために，保存語彙の情報を 1 つのリスト情報として保持するようにした．その際，新規語彙の追加をサポートするため，クエリの情報もリスト情報に含めるようにした．保存語彙の情報からクエリの情報へ関連を持たせ，そのクエリ・時刻で出力した語彙と保存語彙との差分の情報を保持し，関連を持たせた．

さらにタグの伝播のアプローチを行うため，タグの形態素も予めタグ情報とした．形態素解析には MeCab[4] を使用した．タグとして利用する語の読み情報は各形態素の読みから推定するようにした．

5.1 語彙情報の収集

最初に保持すべき基本語彙としては ipadic version 2.7.0 の語や日本郵便郵便番号データベース[5] の地名情報を利用した．表 1 に今回までに使用している語彙の情報源とその中で用いた情報を示す．2008 年 4 月 22 日時点で合計 566545 語のデータを保有するデータベースを構築した．

日本郵便の郵便番号データベース，三菱電機 EPG データは，CSV 形式で与えられ，そこから必要な情報を抽出した．はてなキーワード API[6] は，はてなキーワードのコンテンツを任意のアプリケーションから利用するための API であり，キーワードの読み

表 1: 語彙整備に用いた情報源 (タグ情報: タグに使用した情報．種類は以下の通り：(A) 話題性の高い語の読み，(B) 標準語彙の読み，(C) 地名の読み，(D) 飲食店名の読み，(E) 宿名と温泉名の読み，(F) 人名（ミュージシャン，タレント）の読み，(G) 各タグの読み

情報源	種類	タグ情報
はてなキーワード API	(A)	カテゴリ
Yahoo!辞書 - 新語探検	(A)	カテゴリ
イザ語	(A)	カテゴリ
ipadic version 2.7.0	(B)	品詞
郵便番号データベース	(C)	市区町村 都道府県
ホットペッパー Web サービス	(D)	所在地 ジャンル
ぐるなび Web サービス	(D)	所在地と カテゴリ
じゃらん Web サービス	(E)	所在地
MusicNavi Web サービス	(F)	人名
三菱電機 EPG データ	(F)	人名
Wikipedia	(G)	収集語彙の 記事カテゴリ

とカテゴリ情報を直接 API から取得した．Yahoo!辞書 - 新語探検[7]，イザ語[8]，はいずれもニュースや話題語が得られるサービスであり，HTML を解析し，読みとカテゴリの情報を取得した．ホットペッパー Web サービス[9]，ぐるなび Web サービス[10]，じゃらん Web サービス[11]，MusicNavi Web サービス [5] からは，WebAPI 経由で地名や店名，カテゴリ，人名等をクエリとして語彙情報を得た．また，Wikipedia で対象語を検索し，該当項目があった場合にその関連語のタグ付けを試みた．HTML 内の表や箇条書きを関連語とする手法 [7] があるが，Wikipedia には記事の下部に独立した要素としてカテゴリ情報があるため，HTML を解析し，その情報のみを抽出してタグ付けを行った．その際，カテゴリの語が新規の語であったときはその語も登録し，それに対してもタグ付けを行った．

5.2 語彙リストの作成・保存

ユーザは検索画面上部のフォームにクエリを入力する．クエリは and,or,not や () によって複数の条件を指定できる．クエリの入力後，フォームの下の検索結果出力画面にそのクエリでの検索結果のリス

[4]MeCab: Yet Another Part-of-Speech and Morphological Analyzer,http://mcab.sourceforge.net/

[5]日本郵便，郵便番号データダウンロード,http://www.post.japanpost.jp/zipcode/download.html

[6]はてなキーワード, http://d.hatena.ne.jp/keyword

[7]新語探検，http://dic.yahoo.co.jp/newword

[8]イザ語, http://www.iza.ne.jp/izaword/

[9]ホットペッパー.jp,http://www.hotpepper.jp/

[10]ぐるなび,http://www.gnavi.co.jp/

[11]じゃらん net,http://www.jalan.net/

トが表示される．ユーザはその中で不適切だと思われる語に対して，その名前の左にあるチェックボックスをオンにし，削除ボタンを押して不適切語を除去する．リストの右隣にはリストの名前を入力し，登録するフォームがある．ユーザはここにリストの名前を入力して登録すると各形式へ出力するための保存画面が現れる．希望する形式のボタンを押すと，その形式でリストがダウンロードできる．出力形式はCSVファイルやJulius孤立単語認識用辞書形式などが用意されている．リストを表示した際，各語の右隣には語の詳細情報画面へのリンクと修正ページへのリンクがあり，ユーザはここから語彙のタグ情報や情報源へのリンクを参照したり，1語単位での修正ができる．

ユーザが作成した語彙リストはデータベースに保持され，検索画面右部に生成リスト詳細画面へのリンクが表示される．ユーザはここから過去の語彙リストの作成・更新の履歴が参照できる．

今回作成したプロトタイプシステムでは新規語獲得規則を得るためのクエリの生成は行わず，語彙リスト生成時に用いたクエリを新規語の獲得規則に使用している．語彙リスト生成後に新しくデータベースに追加された語でこのクエリの条件に合致したもののリストを生成リスト詳細画面の下部に表示する．リスト内の不適切語に対してチェックボックスをオンにして除去し，語彙リストを更新することができる．

6 クエリベースによる語彙リストの正確性評価

プロトタイプシステムを用いてクエリによる語彙リスト表現の正確性について評価を行った．

6.1 評価方法

目的を決め，それに対してクエリベースの検索での出力結果の正確性を評価した．その際の評価基準として以下のような適合率，再現率とF値の値を使用した．

$$\text{適合率} = \frac{\text{適合文書数}}{\text{検索結果文書数}}$$

$$\text{再現率} = \frac{\text{適合文書数}}{\text{正解文書数}}$$

$$F\text{値} = \frac{2 \times \text{適合率} \times \text{再現率}}{\text{適合率} + \text{再現率}}$$

ここで適合文書数は出力結果中の目的を満たす語の数，検索結果文書数は出力結果の語の数，正解文書数はデータベース内の目的を満たす全ての語の数である．

目的の語が十分に含まれると思われる検索クエリを決め，その出力結果から目的に該当しない語を人手で取り除く．こうして作られたリストを正解文書と仮定する．このときに使ったクエリを基準クエリ，基準クエリでの出力結果を基準セットと呼ぶ．データベースや基準セットに含まれない目的の語はここでは無視する．

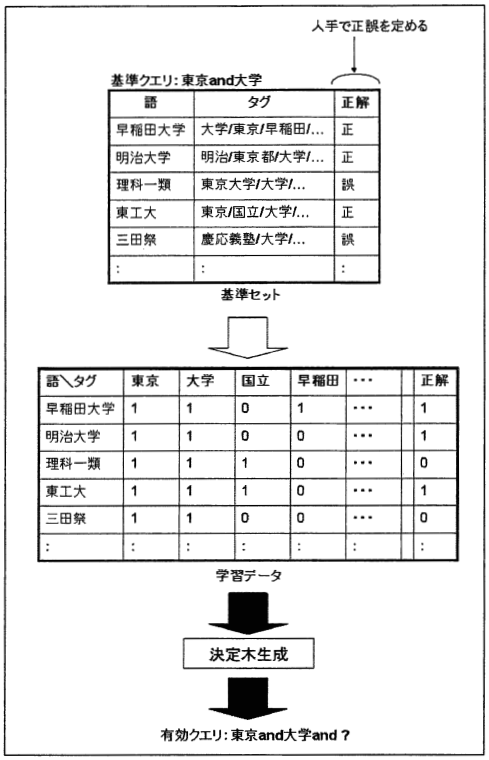

図 3: 有効クエリの決定

正解文書に近くなるようなクエリを“基準クエリに付け加えるクエリ”という条件の下で探す．基準セットの語に付けられた全てのタグに対して，各語にどのタグが付けられているか否かを属性変数として決定木を作成．その上で正解と不正解を分別するのに有効なノードを選び出し，それに割り当てられたタグを用いて正解文書に近くなるようなクエリをいくつか求めた．求めたクエリを有効クエリと呼ぶ(図3)．

6.2 実験

データベースは2007年12月17日時点，総語数526042語のものを用いた．いくつかの目的に対して基準クエリ，有効クエリを指定して実験を行った．決定木の作成にはWeka[12]を用いた．また，タグの細粒語としてタグとなる語の形態素を用いた．形態素解析にはMecabを用いた．

6.3 結果

出力語彙の例を表2に，基準クエリと各有効クエリの結果を表3に示す．いづれの目的においても適当な条件を追加すればF値を改善させることができた．特に2.4の試行では再現率を維持したまま適合率を大きく向上させることができた．

[12]Weka 3 - Data Mining with Open Source Machine Learning Software in Java,http://www.cs.waikato.ac.nz/ml/weka/

表 2: 出力語彙の例 (番号は試行番号, 括弧付きの試行番号は基準クエリを表す.)

番号	クエリ	検出語例	不適切語例	除去された語例
目的: 東京都の大学名のリスト				
[2.1]	東京 and 大学	明治大学/東京都の大学/立教大学/東京農工大学/明治神宮野球場/高橋由伸/東京六大学野球連盟	東京都の大学/明治神宮野球場/高橋由伸/東京六大学野球連盟	
2.2	東京 and 大学 and 固有名詞	明治大学/立教大学/東京農工大学/神宮球場/成城大学/東工大/一文	神宮球場/一文	高橋由伸/東京都の大学/明治神宮..
2.4	東京 and 大学 not 野球	明治大学/東京都の大学/立教大学/東京農工大学/お茶の水女子大学/東京四大学	東京都の大学/東京四大学	高橋由伸/明治神宮../東京六大学野..
目的: 日本の議員名のリスト				
[4.1]	日本 and 議員	鈴木寛/日本の国会議員/羽仁五郎/中曽根康弘/市議会議員/議員連盟	日本の国会議員/市議会議員/議員連盟	
4.3	日本 and 議員 and 国会	鈴木寛/日本の国会議員/羽仁五郎/中曽根康弘/議員連盟	日本の国会議員/議員連盟	市議会議員
4.4	日本 and 議員 not 連盟	鈴木寛/日本の国会議員/羽仁五郎/中曽根康弘/市議会議員	日本の国会議員/市議会議員	議員連盟

7 まとめと今後の予定

音声認識アプリケーション開発の問題として，システムが認識可能な語彙の適切な設計と，実際に利用されている語彙のメンテナンスを挙げ，これらの問題を解決するために，集合知を利用した語彙情報の収集・共有・管理システムを提案した．具体的には，語彙情報を集中管理するためのオンラインデータベースシステムを構築し，それをウェブシステムとして利用者・アプリケーションに公開する．提案システムでは，Web資源からの自動収集の枠組みを備え，アプリケーション用の語彙の新規作成から，その継続的な更新まで包括的な解法を提供する．また，実際に提案システムの実装としてプロトタイプシステムの開発を行い，2008年4月22日時点で合計566545語のデータを保有するデータベースを構築した．また，提案システムを用いて生成可能な語彙リストの正確性に関して評価を行い，提案システムによって実際に有効な語彙リストの生成が可能である事を示した．

今後は情報源の追加，リスト作成機能の強化，有効クエリの提示機能の実装，新規語彙通知機能の実装，作成リスト共有機能の実装などを行う．また，Proxy-Agentとの連携システムについての具体例を示す．本システムは，これらの実装が整い次第，WEBアプリケーションやWEB APIとして公開を予定している．

謝辞 本研究は，経済産業省・平成19,20年度戦略的技術開発委託費「音声認識基盤技術の開発」及び早稲田大学理工学研究所・プロジェクト研究「音声認識基盤技術」の一部として実施されたものである．

参考文献

[1] 古井 貞熙, 他, “音声認識技術実用化に向けた先導研究,” NEDO 平成17年度成果報告書, 100007350, 2006.

[2] Lisa J. Stifelman, Barry Arons, Chris Schmandt, Eric A. Hulteen. VoiceNotes: A Speech Interface for a Hand-Held Voice Notetaker. Proc. of CHI93, pp.179–186, 1993.

[3] 緒方淳, 後藤真孝, 江渡浩一郎, “PodCastle：ポッドキャストをテキストで検索, 閲覧, 編集できるソーシャルアノテーションシステム, ” WISS2006, Dec 2006.

[4] 西村竜一, “音声入力 Web システムを用いた辞書共有型音声認識サービス, ” 日本音響学会 2007 年秋季研究発表会講演論文集, pp61–62.Sep 2007.

[5] 原直, 宮島千代美, 伊藤克亘, 武田一哉, “多様な音響環境下における音声認識システム利用時のデータ収集システム, ” 電子情報通信学会論文誌, Vol.J90-D, No.10, pp.2807–2816, 2007.

[6] Teppei Nakano, Shinya Fujie, and Tetsunori Kobayashi. EXTENSIBLE SPEECH RECOGNITION SYSTEM USING PROXY-AGENT. Proc. of ASRU2007, pp.601-606, December 2007.

[7] 新里圭司, 鳥澤健太郎, “HTML 文書からの単語意味クラスの単純な自動獲得手法,” 情報処理学会論文誌,vol.48,no.6,pp2140–2152.June 2007.

表 3: 各クエリ条件に対しての結果 (番号は試行番号, 括弧付きの試行番号は基準クエリを表す. 時間は検索にかかった時間 [秒])

番号	クエリ	検出語	不適切語	適合率	再現率	F 値	時間
目的: 公園名のリスト							
[1.1]	公園	414	122	0.7053	1.000	0.8272	3.713
1.2	公園 and 地理	284	53	0.8134	0.7911	0.8021	7.971
1.3	公園 not の not 場	344	70	0.7965	0.9384	0.8616	12.11
1.4	公園 and(地理 or(not の))	369	83	0.7751	0.9795	0.8654	16.24
1.5	公園 and(地理 or(not の))not 場	358	73	0.7961	0.9760	0.8769	30.53
目的: 東京都の大学名のリスト							
[2.1]	東京 and 大学	305	52	0.8295	1.000	0.9068	9.064
2.2	東京 and 大学 and 固有名詞	222	4	0.9820	0.8617	0.9179	27.43
2.3	東京 and 大学 not スポーツ	277	27	0.9025	0.9881	0.9434	14.23
2.4	東京 and 大学 not 野球	279	26	0.9068	1.000	0.9511	13.49
2.5	東京 and 大学 and	280	27	0.9036	1.000	0.9493	37.78
2.5	(固有名詞 or(not 野球))						
目的: 大阪府の企業名のリスト							
[3.1]	大阪 and 企業	626	19	0.9696	1.000	0.9846	8.939
3.2	大阪 and 企業 not 依頼	614	15	0.9756	0.9868	0.9812	12.90
3.3	大阪 and 企業 not 西	625	18	0.9712	1.000	0.9854	13.10
3.4	大阪 and 企業 not 打線	625	18	0.9712	1.000	0.9854	12.78
3.5	大阪 and 企業 not 西 not 打線	624	17	0.9728	1.000	0.9862	16.08
目的: 日本の議員名のリスト							
[4.1]	日本 and 議員	749	22	0.9706	1.000	0.9851	11.36
4.2	日本 and 議員 not 日本国憲法	745	18	0.9745	1.000	0.9871	14.21
4.3	日本 and 議員 and 国会	740	16	0.9784	0.9959	0.9870	13.98
4.4	日本 and 議員 not 連盟	746	19	0.9758	1.000	0.9871	13.95
4.5	日本 and 議員 and 国会 not 連盟	738	14	0.9810	0.9959	0.9884	19.27
目的: テレビ番組名のリスト							
[5.1]	テレビ and 番組	1914	218	0.8861	1.000	0.9396	9.375
5.2	テレビ and 番組 not ホテル	1826	160	0.9124	0.9823	0.9461	13.17
5.3	テレビ and 番組 not リゾート	1826	160	0.9124	0.9823	0.9461	11.76
5.4	テレビ and 番組 not 施設	1825	160	0.9123	0.9817	0.9458	11.84
5.5	テレビ and 番組 not 制度	1825	159	0.9129	0.9823	0.9463	11.62