汎用シソーラス探索ライブラリの開発

清田陽司 ¶† 阿辺川武 ¶‡ 吉田稔 ¶† 田村悟之 § 坂井哲 § 増田英孝 §

¶NLP 若手の会 (YANS)
http://yans.anlp.jp/

† 東京大学情報基盤センター 図書館電子化研究部門
{kiyota,mino}@r.dl.itc.u-tokyo.ac.jp

‡ 東京大学大学院教育学研究科
abekawa@p.u-tokyo.ac.jp

§ 東京電機大学大学院工学研究科 情報メディア学専攻
{tamura,sakai}@cdl.im.dendai.ac.jp, masuda@im.dendai.ac.jp

1　はじめに

グラフ構造を持った言語資源は自然言語処理研究のあらゆる場面で活用されている。代表的な例としては、NTT 日本語語彙大系などの木構造をもつシソーラスを用いた語の類似度計算などがある。さらに最近は Wikipedia など、ページ間の構造が巨大なグラフ構造をなしていて一種のシソーラス[1]として扱える言語資源が登場し、さまざまなアプリケーションで活用されるようになってきている。その結果、自然言語処理研究においてはさまざまな種類のシソーラスを使いこなすことが必要とされる状況が生まれている。

シソーラスを扱うためのコードは、自然言語処理の研究者が各自書いていることが多い。言い換えると、個々の研究者が「車輪の再発明」をしている状況である。もちろん、グラフ構造の扱い自体を研究対象とする場合などは一からコードを書くことが必要とされる。しかし、さまざまな自然言語処理アプリケーション (e.g. 語義曖昧性解消、情報検索、機械翻訳) においてシソーラスを利用する場合にまでコードを一から書く必要に迫られることは、自然言語処理アプリケーションの研究を始めるハードルを過度に高くしている可能性がある。このハードルを下げるためには、典型的なシソーラス処理をパッケージ化した共通基盤としてのライブラリが存在することが理想である。しかし、現時点では広く普及しているライブラリは存在しない。

我々は自然言語処理研究のハードルを低くし、研究分野の裾野を広げることを目指して、頻繁に利用されるシソーラス探索処理をパッケージ化したライブラリの可能性についての議論を行った。まず、これまでシソーラスを扱うライブラリが普及しなかった原因を以下のように整理した。

- **プログラミング言語の壁**
  研究に利用するプログラミング言語は、用途や個人の好み、研究組織ごとに蓄積されているライブラリなどによって左右されるため、標準言語を定めることは難しい。また、プログラミング言語の壁を超えて汎用的に利用されるグラフ探索アルゴリズム実装は存在しなかった。

- **データ構造の標準階層モデルが存在しない**
  汎用的なデータ構造を実現するには、機能に応じて階層的に分割された標準的なモデル (cf. データ通信における OSI 基本参照モデル[2]) があることが望ましいが、シソーラスのデータ構造についてそのようなモデルは現在のところ存在しない。

- **シソーラスデータ流通に伴う問題**
  シソーラスの構築には多大なコストがかかるため、有料で入手する必要があるものも少なくない。また、シソーラスデータにアクセス用ライブラリが添付されていることはほとんどない。それらのシソーラスへのアクセス用ライブラリを利用者がボランティアで開発・公開しているケースもあるが、様々なシソーラスを網羅できるようなライブラリを一利用者が開発することは、検証に必要なデータの入手にかかるコストを考慮すると著しく困難である。

- **研究組織の壁**
  自然言語処理分野に携わる研究組織は国内外に散在するが、各組織内で機能的に類似したライブラリ群をそれぞれ開発・共有しているにもかかわらず、組織外との共有の取り組みはあまり活発ではなかった。

[1]「シソーラス」という用語は、本来は「言葉を意味上の類似関係、包含関係などによって構造化した辞書」という意味で用いられるが、本稿では広く「個々のノードに文字列が付与されたグラフ構造」という意味で用いている。

[2]コンピュータの持つべき通信機能を階層構造に分割したモデル。物理層・データリンク層・ネットワーク層・トランスポート層・セッション層・プレゼンテーション層・アプリケーション層の 7 階層で構成される。TCP/IP などの通信プロトコルは OSI 基本参照モデルに対応しているため、高い汎用性をもっている。

さらに、これらの原因を解消し、広く利用されるライブラリを実装するための方法論を以下のようにまとめた。

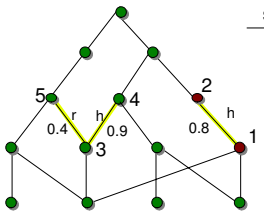
- 基本機能は C++言語で実装し、他のプログラミング言語へのラッパーとして SWIG[3]を採用する。これによって、自然言語処理分野の研究者に利用されている Perl, Ruby, Java, Python などの諸言語に対応する。また、汎用的なグラフ探索アルゴリズム実装である The Boost Graph Library[4]を採用する。これによって、必要なグラフ探索アルゴリズムを利用できるようにする。グラフ探索にはエッジに付与された重みを考慮する。
- シソーラスのデータ構造を、redirect, leaf, node の 3 階層 (後述) からなるグラフとして表現する。これによって、多様なシソーラスに対応するための柔軟性を確保する。また、グラフのノードには個別のシソーラス特有の属性を付与できるようにする。
- 大部分のシソーラスは知的財産として扱われるためライブラリと同時に配布することはできないが、主要なシソーラスについては変換スクリプトを用意しておくことで、シソーラスのデータファイルさえ入手すれば容易にライブラリを利用できるように配慮する。
- NLP 若手の会の枠組みを活用し、開発者間で定期的に実装についてのディスカッションを行う。また、継続的にライブラリについての意見や要望をとりまとめ、実装に反映させていく。

現在、この方針にもとづいて汎用シソーラス探索ライブラリの試作を行っており、WordNet、Wikipedia、件名標目表などのグラフ探索ができるようになっている。

本稿の構成を以下に述べる。2 節において、汎用シソーラス探索ライブラリにおけるシソーラスデータ構造と、現実のシソーラスとのマッピングについて述べ、3 節では具体的な実装方法と、ライブラリに実装するインターフェース (API) について述べる。4 節で本ライブラリの応用例を示し、5 節でまとめを述べる。

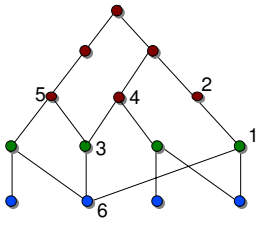
2 データ構造

汎用シソーラス探索ライブラリのデータ構造は、各種シソーラスのグラフ構造を汎用的に表現できるように、ノード間の関係を記した **edge ファイル**、ノードそのものの情報を記した **node ファイル**に分割されている。それぞれのファイルは、タブ区切り形式のテキストファイル (TSV 形式) でライブラリの利用者が各自準備する (主要なシソーラスについてはライブラリ附属のスクリプトにより生成できるようにする)。ライブラリは、edge ファイル・node ファイルを事前に読み込み、リレーショナル型データベースに格納する。



source	target	weight	relation
1	2	0.8	h(hypernym)
3	4	0.9	h(hypernym)
3	5	0.4	r(relative)
:	:	:	:

図 1: edge ファイル



id	name	kind
1	eagle	leaf
2	bird	node
3	cat	leaf
4	mammal	node
5	pet	node
6	kitten	redirect
:	:	:

図 2: node ファイル

2.1 edge ファイル

edge ファイルは、1 レコード (1 行) につき 2〜4 個のフィールド `source`, `target`, `[weight]`, `[relation]` から構成される TSV 形式ファイルである。具体例を図 1 に示す。

`source`, `target` にはグラフ構造中の各エッジの始点ノード、終点ノードの固有 ID(後述) が入る。上位・下位関係をもつグラフ構造を表現する際には、`source` に子ノード、`target` に親ノードの ID を格納することとする。`weight`, `relation` は任意のフィールドである。`weight` にはエッジの重み (浮動小数点形式)、`relation` にはエッジの種別[5]が入る。

2.2 node ファイル

node ファイルは、1 レコード (1 行) につき 3 個以上のフィールド `id`, `name`, `kind`, `[prop_1, ..., prop_n]` から構成される TSV 形式ファイルである。具体例を図 2 に示す。

`id` には各ノードの固有 ID(正の整数値) が入る。`name` には各ノードの名称 (文字列) が入る。`kind` にはノードの種類を示す文字列 (後述) が入る。`[prop_1, ..., prop_n]` は用途に応じて自由に属性を格納することができるフィールドである。

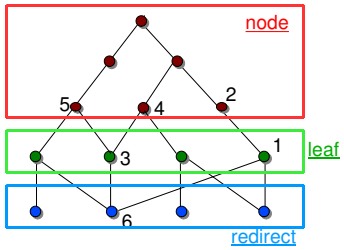
[3] `http://www.swig.org/`, Simplified Wrapper and Interface Generator

[4] `http://www.boost.org/libs/graph/doc/`

[5] 例えば WordNet では hypernym, holonym, entailment などの種別が格納される。

表 1: 各種シソーラスにおけるマッピング

シソーラス	edge ファイル	node ファイル
NTT 日本語語彙大系, EDR, 分類語彙表	概念間関係, 単語→概念	単語 (leaf), 概念 (node)
WordNet	synset 間関係 (同意, 反義, ...), 単語→ synset	単語 (leaf), synset(node)
Wikipedia	カテゴリ間関係, 項目→カテゴリ, リダイレクト・曖昧性回避→項目	リダイレクト、曖昧性回避 (redirect), 項目 (leaf), カテゴリ (node)
図書件名標目表 (LCSH, BSH), MeSH	件名間の関係 (上位下位、関連語など)	件名 (node)
2 言語間辞書	source 言語の単語 → target 言語の単語	source 言語の単語 (redirect), target 言語の単語 (leaf)
表記揺れ辞書	表記揺れを含む単語 →代表単語	表記揺れを含む単語 (redirect), 代表単語 (leaf)



id	name	kind
1	eagle	leaf
2	bird	node
3	cat	leaf
4	mammal	node
5	pet	node
6	kitten	redirect
:	:	:

図 3: 3 階層モデル

2.3 各種シソーラスにおけるマッピング

実際の各種シソーラスを edge ファイル、node ファイルにあてはめた例を表 1 に示す。「node ファイル」欄において、括弧内に示している文字列は `kind`(ノードの種類) を意味している。

ここに挙げたシソーラスは、すべて図 3 に示すように `redirect`, `leaf`, `node` の 3 階層の組み合わせで構成されている (いずれかの階層が存在しないものもある)。Wikipedia を例にとって説明すると、`redirect` は代表表記に汎化されるべきキーワード (例: 東大)、`leaf` は個々の項目 (例: 東京大学)、`node` は個々の項目を束ねるカテゴリ (例: 日本の国立大学、旧帝国大学、...) である。`leaf` → `node`, `node` → `node` は下位概念→上位概念のエッジとしてとらえることができる。NTT 日本語語彙大系などでは個々の `node` はただ 1 個の上位概念しかもたないが、Wikipedia のように個々の `node` が複数の上位概念をもつものもある。

3 実装と API

汎用シソーラスライブラリの中核部分は、The Boost Graph Library を利用するため C++言語で実装されている。また、リレーショナル型データベースには利用の際の簡便性を考慮して組み込み型の SQLite を採用している。なお、サーバ・クライアント型データベース (MySQL など) への対応も検討している。

グラフ探索系 API
- 木構造グラフ用
  - `get_parent` (ノードの親ノード)
  - `get_child` (ノードの子ノード)
  - `get_depth` (ルートノードからの深さ)
  - `calc_similarity` (2 ノード間の類似度)
  - `print_tree` (ノードツリーの表示)
- 汎用
  - `get_shortest_path` (2 ノード間の最短パス) (breadth first, dijkstra)
  - `get_distance` (2 ノード間の距離)

ノード参照系 API
- ノードの情報取得
  - `get_name` (ノードの名称)
  - `get_kind` (ノードの種類 redirect/leaf/node)
  - `is_redirect`, `is_leaf`, `is_node` (ノードの種類を true/false で判別)
- ノード名の検索
  - `exact_match` (完全一致)
  - `search_forward` (前方一致)
  - `search_backward` (後方一致)

図 4: 汎用シソーラス探索ライブラリの主な API

ライブラリの関数群 (API, Application Programming Interface) は、**グラフ探索系 API** と**ノード参照系 API** に大きく分けられる。実装済みの主な API を図 4 に示す。これらの API は、The Boost Graph Library を利用して実装されている。API を活用することにより、広範な種類のシソーラスを活用したアプリケーションを容易に構築することが可能になる。

また、SWIG を利用して Perl, Ruby, Java, Python などへのラッパーを実装している。これによって、C++言語に馴染みのない方にも広く利用していただけることを目指している。

4 応用例

汎用シソーラス探索ライブラリは、単一のシソーラスを利用したアプリケーションの構築を容易にすることはもちろん、複数のシソーラスを組み合わせたアプリケーションを実装することにも活用が可能である。

汎用シソーラスライブラリを活用して実装可能なアプリケーションの例を以下に示す。

和英辞書と WordNet の結合

WordNet は、世界中で広く用いられているシソーラスであり、各言語版とその間のリンクの開発や、単語類似度測定等の応用について、これまでに多くの研究がなされている。和英辞書を用いて日本語の単語を WordNet 上にマッピングし、さらに、WordNet 上で得られた単語 (synset 中の単語等) を英和辞書を用いて日本語の単語に戻すことにより、WordNet に関する豊富な研究成果を、日本語を扱う場合にも利用することが可能になる。

例えば、WordNet 類似度 [1] を介した日本語の単語同士の類似度計測や、WordNet を利用した高度なテキスト検索 [2] の日本語文書検索への適用等が考えられる。日本語シソーラス単独で行えるタスクについても、これら WordNet を介することによって得られる結果は、結果の補完等に利用できるという点で有益であると考えられる。

Wikipedia と MeSH の結合

生命科学分野のシソーラスである MeSH (Medical Subject Headings) は、医学分野の文献を検索する際に標準的に利用されている。しかし、MeSH に含まれる専門用語に精通していないと利用することは難しい。専門外の人が自身の病気の治療法について探す場合などは、一般に使われていることばで検索できることが有用だと考えられる。

Wikipedia 日本語版、英語版、MeSH を組み合わせて利用することで、MeSH を一般に使われている用語で検索できるアプリケーションが実現できるかもしれない。例えば、「床ずれ」は Wikipedia 日本語版にて「褥瘡 (じょくそう)」にリダイレクトされていて、「褥瘡」は Wikipedia 英語版「Bedsore」へのリンクを持っている。さらに「Bedsore」は MeSH にて件名「Pressure Ulcer」に関連づけられている。MEDLINE などの論文データベースで「Pressure Ulcer」を検索することで、床ずれの症例・治療法に関する文献を見つけることができる。

Wikipedia と件名標目表の結合

図書館で資料を効率的に探すには、探している分野に対応する書架分類記号 (日本十進分類法など) を知る必要があるが、専門外の人にとっては難しい。書架分類記号を探すためのツールとして件名標目表 (基本件名標目表 (BSH)、国立国会図書館件名標目表 (NDLSH) など) が存在するが、日本語の件名標目表については語彙数が十分でなく、必ずしも有用とはいえない。

Wikipedia カテゴリのグラフ構造を幅優先探索し、件名標目表中の文字列レベルで一致する標目を抽出することで、Wikipedia に含まれるあらゆる項目名に対して件名、書架分類記号を提示することができる [3]。また、この方法で導出された部分グラフ構造を描画して利用者に表示することで、利用者の情報要求の具体化に役立てることも可能である [4]。

5 おわりに

本稿では、多様なシソーラスを統合的に利用できる汎用ライブラリ実装の取り組みを紹介し、いくつかのアプリケーションを示した。現在、ライブラリの一般公開に向けて詳細な仕様を検討中である。多くの研究者に利用されることを目標とし、本発表のデモに対するフィードバックを生かした上で一般公開する予定である。また、ライブラリの実装に協力していただける方を歓迎します。

謝辞

本研究は、文部科学省科学研究費補助金若手研究 (B)(課題番号 18700134) の助成を受けて遂行されました。また、本研究の基本的なアイディアは、NLP 若手の会 (YANS) の合宿でのディスカッションから生まれました。貴重な意見をいただいた NLP 若手の会有志のみなさまに感謝いたします。

参考文献

[1] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::similarity - measuring the relatedness of concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, 2004.

[2] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. The MIT Press, 1998.

[3] 清田陽司, 田村悟之, 中川裕志, 増田英孝. Reference Navigator： 異種オントロジーの統合ブラウジングツール〜図書館の分類体系と Wikipedia カテゴリの対応付け〜. 言語処理学会 第 13 回年次大会ワークショップ「言語的オントロジーの構築・連携・利用」論文集, pp. 35–38, 2007.

[4] 坂井哲, 清田陽司, 増田英孝, 中川裕志. 図書館と Web の分類体系を統合的に活用したテーマグラフ可視化インタフェース. 情報処理学会 第 70 回全国大会 講演論文集, 2008. (to appear).