Vol.2013-NL-214 No.10
2013/11/15

フレーズベースTF-IDF: 名詞句解析の応用

村脇 有吾[1,a)]

概要：文書中の重要語の認識は様々な応用の基礎となるタスクである．そうした重要語は，しばしば単語ではなく，単語列からなる．しかし，教師なし手法における state-of-the-art は，単語 TF-IDF の総和によるスコア付けであり，単語列の意味的まとまりを認識しない．そこで，本稿では，名詞句の内部構造解析を応用し，複数の単語からなるフレーズに対して直接 TF-IDF を算出する手法を提案するとともに，その振る舞いを調べる．

キーワード：キーフレーズ抽出，TF-IDF，名詞句解析

1. はじめに

自然言語処理の様々なタスクにおいて，文書を bag-of-words によって近似する手法が強力なベースラインとして用いられている．しかし，複雑な概念は，しばしば単語ではなく，単語列によって表現される．したがって，人間に提示するテキスト要約表現としては，単語よりも，一般に複数の単語からなるフレーズの方が適切である．

そのようなフレーズの認識自体をタスクとするものに，キーフレーズ抽出がある．このタスクでは，文書集合を入力とし，各文書を代表するキーフレーズを出力する．キーフレーズとしては，本稿では，その大半を占める名詞句に対象をしぼる．

フレーズを扱う際，単語の場合にはない 2 つの問題が生じ得る．ひとつは，“new optimal control problems” から “new optimal control” を抽出するといったように，文法的に不適切な単語列を抽出し得るという問題である．もう一つは，“optimal control problems” に対する “new optimal control problems” のように，あまり意味的にまとまりのないフレーズを抽出し得るという問題である．本稿では，それぞれを文法性，語彙性の問題とよぶ．

キーフレーズ抽出への取り組み方は，教師あり手法 [5], [30]，大規模外部知識を用いる手法 [28]，教師なし手法に大別できる．訓練データや，対象分野を被覆する外部知識が得られる場合には，教師あり手法や外部知識を用いる手法が現実的な選択肢である．しかし，本稿では，キーフレーズの振る舞いを直接捉えることを目的に，教師なし手法を採用する．また，教師なし手法には，教師データや外部知識が存在しない新たな分野への適用が容易という利点もある．

教師なしキーフレーズ抽出の手法としては，グラフに基づく順位付け [20], [32] やクラスタリング [18] が用いられてきた．しかし，[7] は，こうした複雑な手法が，TF-IDF に基づく単純な手法（単語ベース TF-IDF 法）にほぼ一貫して負けることを示した．

そこで，本稿では，単語ベース TF-IDF 法をベースラインとみなし，そのスコア付けを再定式化する．従来研究は，単語ベース TF-IDF 法が構成単語の TF-IDF スコアの和であるという点のみに着目してきた．しかし，もう一つ重要な点として，最長名詞句に基づくヒューリスティクスを用いて候補を絞り込むことにより，文法性の問題を回避していることを改めて示す．次に，単語ベース TF-IDF 法の経験的振る舞いを調査し，このヒューリスティクスでは，特に長い文書について，語彙性の問題が解消できていないことを示す．

単語ベース TF-IDF 法がキーフレーズ候補を単語に分解するのに対し，本稿では，複数の単語からなるフレーズを一体として認識する手法を模索する．すなわち，フレーズに対して直接 TF と IDF を算出する手法，フレーズベース TF-IDF 法を提案する．フレーズベース TF では，ヒューリスティクスによりキーフレーズ候補を絞り込むことなしに文法性の問題を解消するために，名詞句の内部構造解析を利用する．また，語彙性の問題もフレーズベース TF による解消が期待できる．フレーズベース IDF では，文法

[1] 九州大学
Kyushu University
[a)] murawaki@ait.kyushu-u.ac.jp

性，語彙性の問題について特に対策を行わない．本稿では，実験を通してフレーズベース TF-IDF の経験的な振る舞いを調べる．

2. 関連研究

確率的言語モデルは多くのタスクで有効性が示されており，キーフレーズ抽出に対しても有効と期待されるかもしれない．しかし実際に採用した例は少ない．推測される理由としては，確率値は解釈が難しいことが挙げられる．確率的言語モデルは長い単語列に極端に小さな値を与える．そのため，例えば候補をフィルタリングするために適当な閾値を設けるといったことが難しい．[27] は，確率値を単独で使うのではなく，2 つのコーパスから算出された確率値を対照させている．しかし，小さな値同士の割り算は不安定な振る舞いをすると推測される．

キーフレーズ抽出と関連するタスクとして用語抽出（term extraction）がある [12], [22]．このタスクでは，コーパスを入力とし，コーパス全体を代表する用語を抽出する．これに対し，本稿が対象とするのは，個々の文書を代表するキーフレーズであり，コーパス全体では頻出しない候補も適切に扱いたい．

トピックモデリングの分野では，LDA（latent Dirichlet allocation）の拡張として，コロケーションのモデル化が行われている [6], [33]．いずれもフレーズをバイグラムの連鎖に分解する．[17] は階層 Pitman-Yor 過程を用いて，バイグラムを N-gram に拡張している．[10] は，Adaptor Grammar とよばれる確率的文脈自由文法の拡張を用いて，複数の単語からなるフレーズを一体として認識できるトピックモデルを提案している．ただし，Adaptor Grammar には，推論を容易にするためにパラメータを積分消去する場合，自己再帰を正しく扱えないという難点が知られている [2]．したがって，このモデルの単純な拡張では，フレーズを入れ子にすることはできない．また，これらの研究はいずれも，得られたフレーズが適切なまとまりであったかを実験を通じて検証していない．

3. 実験設定

3.1 データセット

キーフレーズ抽出のデータセットを 2 個用いる．一方は長い文書の代表例として，もう一方は短い文書の例とする [*1]．

3.1.1 NUS

NUS キーフレーズコーパス [24] は，科学に関する英語の会議論文 211 本からなる．実験ではそのすべてを用いる．各文書に対して，著者および複数のアノテータがキーフレーズを付与している．[7] にならい，それらの和集合を正解データとみなす．

各文書は 4 から 12 ページ，語数にして平均 8,187 語からなる．文書ごとの正解キーフレーズ数は平均 11.0 であり，正解キーフレーズの語数は平均で 2.1 語である．キーフレーズ候補が文字通り数千にのぼるのに対し，正解は 10 個程度に過ぎず，高い精度を出すのが難しいデータセットとなっている．

3.1.2 Inspec

Inspec コーパス [8] は，英語のジャーナル論文 2,000 本からなる．各論文は，標題，要旨およびキーフレーズ一覧からなる．標題と要旨をあわせて文書とみなす．各文書には，統制されたキーフレーズと統制されていないキーフレーズが付与されている．前者は，あらかじめ定義されたシソーラスによって統制されているが，後者は自由に付与されている．[7] にならい，実験では統制されていないキーフレーズを正解とみなす．正解キーフレーズのなかには，文書中に一度も出現しないものが含まれてい るが，特にフィルタリングは行わない．

[8] は 2,000 文書のコーパスを 3 セットに分割し，1,000 を訓練，500 を確認，500 を評価に用いている．本実験では，500 文書の評価セットを用いる．評価セットでは，各文書は平均 134 語からなる．これは NUS のわずか 1.6%にすぎない．文書あたりのキーフレーズ数は平均 9.8 で，キーフレーズの語数は平均 2.3 語である．

3.2 前処理

キーフレーズ抽出の前処理として以下を行う．まずヒューリスティックな規則を用いて各文書を文に分割する．次に文のトークン化と品詞タグ付けを Lookahead POS Tagger[29] を用いて行う．このタガーの訓練には，Penn Treebank[19] の Wall Street Journal（WSJ）部分と Brown Corpus 部分を用いた．正確には，論文に頻出する “[” と “]” をそれぞれ開き括弧および閉じ括弧と認識できるようにするために，さらに 5 文を訓練データに追加した．

3.3 名詞句チャンキング

前処理された各文書から名詞句を抽出し，それらをキーフレーズ候補とする．ここで，ストップワード等は用いない．

名詞句抽出において，[7] は，[32] と同様に，品詞タグに基づく規則を用いている．具体的には，彼らは以下の条件すべてを満たす単語列を抽出している．

- 各単語に Penn Treebank の品詞タグで `NN`，`NNS`，`NNP`，`NNPS` もしくは `JJ`（名詞あるいは形容詞）が付与されている．
- 最後の単語が名詞である．

このように対象を名詞と形容詞に絞り込むのは良い近似

[*1] [7] は 4 種類のデータセットを用いた調査結果を報告しているが，複数の手法の振る舞いを見る限り，長い文書と短い文書の大きく 2 種類に分けられると判断した．

であるが，再現率の上界を低下させる．例えば，この手法では名詞句 “computing system” の抽出に失敗する．なぜなら “computing” はよく分詞 VBG としてタグ付けされるからである．

より言語的に自然なまとまりを抽出することを目的に，本稿では名詞句チャンキングを用いる．[*2] 具体的には，CRF++[*3] を用いてチャンカを実装する．訓練には CoNLL-2000 shared task[26] で提供されたデータセットを用いる．予備実験では，訓練データで訓練し，テストデータで評価したとき，名詞句（NP）の F 値は 94.19%となった．以降で用いるモデルは，訓練データとテストデータの両方を用いて訓練する．

CoNLL-2000 の定める名詞句と本稿で対象とする名詞句の間には若干の齟齬がある．例えば，代名詞や先頭の冠詞は本稿では不要である．この問題に対処するため，以下の規則を順に適用するという後処理を行う [*4]．

(1) チャンクを等位接続の CC や “,” でサブチャンクに分割する．ただしこれらの区切りはサブチャンクに含まない．等位接続の適切な処理は今後の課題とする．[*5]
(2) チャンクが PRP，WDT，WP あるいは EX（代名詞等）を含む場合は破棄する．
(3) 単語列を走査して最右の DT，PRP$，WP$，WRB，PDT，CC，POS，(あるいは``を探す．見つかった場合は，この区切りを含む左側単語列を取り除く．
(4) 括弧と引用符を取り除く．

これらの操作によって得られた名詞句の各出現を**最長名詞句**とよぶ．文書 doc に対して，最長名詞句を集めて構成した名詞句集合を $\mathrm{longest}(doc)$ で表す．

本稿では，最長名詞句だけでなく，その部分列（**部分名詞句**）もキーフレーズ候補とする．ただし，最後の単語が名詞の場合に候補を限定する．こうして拡張された名詞句集合を $\mathrm{all}(doc)$ で表す．

3.4 評価尺度

いくつもの従来研究が指摘するように [14], [24]，キーフレーズは主観的であり評価が難しい．本稿では，簡単のために評価には完全一致を用いる．候補の正規化は小文字化のみを行い，ステミング等は行わない．複数の文書の再現率，適合率，F 値の集約にはマイクロ平均を用いる．

[7] にならい，キーフレーズ抽出の性能を再現率・適合率曲線で報告する．再現率・適合率曲線はシステムの出力量を変化させることで生成する．各システムはキーフレーズ候補を順序付けし，上位 K 候補を出力する．この K を変化させることで出力量を制御する．

4. ベースライン手法

4.1 単語ベース TF-IDF とその変種

教師なしキーフレーズにおいて，単語ベース TF-IDF 法はキーフレーズ候補に対して，構成単語の TF-IDF スコアの和を与える．[7] は，より複雑な他手法とくらべて，この手法がほぼ一貫して精度で上回ることを示した．このため，本稿では，単語ベース TF-IDF 法をベースラインとし，まずはこの手法で得られる順序付けされたキーフレーズ候補を調べる．

最初に単語ベース TF-IDF 法を再定式化する．単語ベース TF-IDF 法では，文書 doc におけるキーフレーズ候補 $\boldsymbol{w} = w_1, \cdots, w_N$ のスコアは以下で与えられる．

$$\mathrm{tfidf}_{doc}(\boldsymbol{w}) = \mathrm{unit}_{doc}(\boldsymbol{w}) \times \mathrm{term}_{doc}(\boldsymbol{w}),$$
$$\mathrm{unit}_{doc}(\boldsymbol{w}) = I(\boldsymbol{w} \in \mathrm{longest}(doc)),$$
$$\mathrm{term}_{doc}(\boldsymbol{w}) = \sum_{i=1}^{N} \mathrm{tfidfW}_{doc}(w_i),$$
$$\mathrm{tfidfW}_{doc}(w_i) = \mathrm{tf}_{doc}(w_i) \times \log(D/D_{w_i}),$$

ここで，$I(statement)$ は $statement$ が真のとき 1，そうでなければ 0 を返す．また，スコアが 0 の候補は出力から除外されるとする．$\mathrm{tf}_{doc}(w)$ は doc における w の頻度，D はデータセット中の文書数，D_w は w が少なくとも 1 回出現する文書数を表す．unit および term という名称は，それぞれ [11] の *unithood* および *termhood* という概念から借用している．unithood は “the degree of strength or stability of syntagmatic combinations or collocations” を表す．一方，termhood は “the degree that a linguistic unit is related to (or more straightforwardly, represents) domain-specific concepts” を表す．

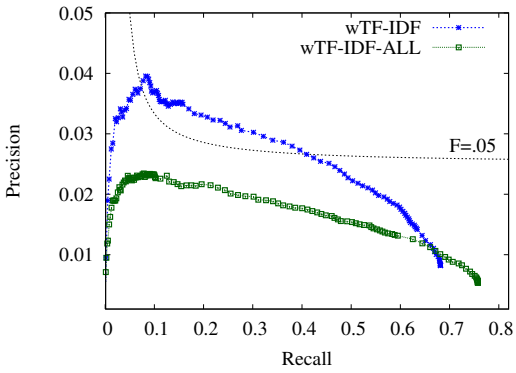
この再定式化は，単語 TF-IDF の和が termhood のみを表していることを示している．unithood は，該当候補が少なくとも 1 回最長名詞句として出現したか否かによってヒューリスティックに判定されている．名詞句チャンキングが高精度と仮定すると，$\mathrm{unit}_{doc}(\boldsymbol{w})$ は文法的に不適格な候補を効果的に出力から取り除く．しかし，部分文字列としてしか出現しない候補すべてが文法的に不適格ではなく，そのうちの一部はキーフレーズである．

$\mathrm{unit}_{doc}(\boldsymbol{w})$ の効果を調べるために，単語ベース TF-IDF 法の変種，単語ベース TF-IDF-ALL 法を考える:

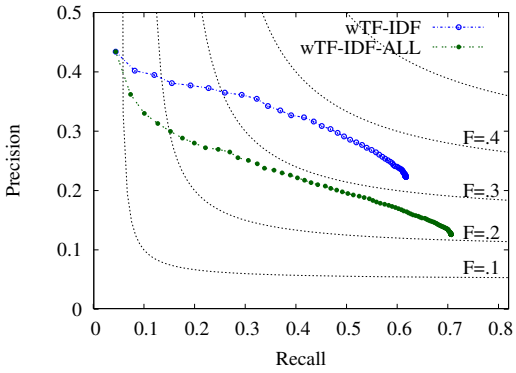
$$\mathrm{tfidfall}_{doc}(\boldsymbol{w}) = I(\boldsymbol{w} \in \mathrm{all}(doc)) \times \mathrm{term}_{doc}(\boldsymbol{w}),$$

単語ベース TF-IDF 法との違いは，$\mathrm{longest}(doc)$ が $\mathrm{all}(doc)$ で置き換えられていることである．

[*2] [8] は品詞タグ規則がチャンキング手法を精度で大幅に上回ったと報告している．しかし，この報告は再現率が非常に低いという点で本稿のチャンカと異なっており，参考にならない．
[*3] http://crfpp.googlecode.com/svn/trunk/doc/index.html
[*4] より自然な解決方法は，本稿の基準にしたがった正解データを作り，それでチャンカを訓練するというものである．今後の課題としたい．
[*5] 本稿では名詞句解析を適用するが，そこでは主辞後置性を仮定している．しかし，英語の係り受け解析では，伝統的に最左の等位項を主辞としており，本稿の仮定にしたがわない [3], [9], [34].



(a) NUS.



(b) Inspec.

図 1: 単語ベース TF-IDF 法と TF-IDF-ALL 法の比較
Fig. 1 Comparison between word-based TF-IDF and TF-IDF-ALL.

4.2 結果と議論

図 1 に単語ベース TF-IDF 法（wTF-IDF）と単語ベース TF-IDF-ALL 法（wTF-IDF-ALL）の比較結果を示す．ここで，NUS については描画の都合上，いくつかの点を間引いている．単語ベース TF-IDF-ALL 法は再現率の上界を NUS について 7.6%，Inspec について 9.0%向上させた．その代わりに，出力候補の総数はそれぞれ 70%，103%増加した．結果として，単語ベース TF-IDF-ALL 法は，全体的な精度を大幅に悪化させた．予想される通り，部分名詞句を効果的に活用するには，文法的に不適格な候補への対策が必要となる．本稿では，この問題を**文法性**とよび，unithood の 1 要素と考える．

単語ベース TF-IDF 法のヒューリスティックな unithood 尺度も，実際には完全からはほど遠い．NUS について，出力が少量の区間（$K < 23$）では，再現率と同時に適合率が向上している．すなわち，単語ベース TF-IDF 法が最上位とする候補は，それに続く候補よりも誤りの割合が大きい．本稿では，この現象を競合候補とよぶ概念で説明する．あるキーフレーズ候補が別のキーフレーズ候補に包含されているとき，キーフレーズ候補のペアが競合している．例えば，“computable bipartite graph” は “bipartite graph” と競合している．競合が発生したとき，$\mathrm{term}_{doc}(\boldsymbol{w})$ は常

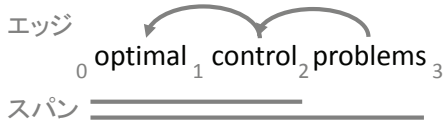


図 2: 名詞句の内部構造
Fig. 2 Internal structure of a noun phrase.

により長い候補に大きなスコアを与える．そのため，単語ベース TF-IDF 法で最上位となる候補は長いフレーズである．しかし，そうした候補はより短い候補とくらべて必ずしも正解の割合が多くない．むしろ正解キーフレーズに余分な要素が付加された候補が現れる．文書が長いほど，こうした誤った候補が出現する機会が増える．

この問題を解決するためには，**語彙性**と本稿がよぶ問題に取り組む必要がある．語彙性もまた unithood の 1 要素である．例えば，“computable bipartite graph” は**構文的**名詞句であり，“computable” が**語彙的**名詞句 “bipartite graph” にその場で付加されている．注意を要するのは，そうした付加要素が必ずしも修飾要素とは限らないことである．例えば，“Round Robin polling strategy” から “strategy” を取り除きたい場合がある．もちろん，語彙的名詞句と構文的名詞句に明確な境界があるわけではない．語彙性は連続的な尺度で表現するのが適当であろう．

5. フレーズベース TF-IDF

単語ベース TF-IDF 法がキーフレーズ候補を単語に分解するのに対し，本稿では，複数の単語からなるフレーズを一体として認識する手法，フレーズベース TF-IDF 法を提案する．フレーズベース TF-IDF 法は，フレーズに対して直接 TF-IDF を算出する．単語ベース TF-IDF 法と同じく，最長名詞句に基づくヒューリスティクスを用いる．ただし，部分名詞句を効果的に活用するために，名詞句解析を利用する．そこで，まず名詞句解析の説明からはじめる．

5.1 名詞句解析

名詞句解析 [1], [15], [16], [23], [25] は，名詞句の内部構造を解析するタスクである．本稿では，エッジとスパンを特徴量として用いるモデル [21] を採用する．名詞句の内部構造は**図 2** のように，エッジあるいはスパンによって表現できる．ここで，エッジは単語ペア間の係り受け関係を表す．一方，スパンは意味的にまとまった部分列を表す．主辞後置性を仮定すると，スパンによる表現は等価なエッジによる表現，すなわち係り受け木に変換できる．したがって，名詞句解析は係り受け解析に帰着できる．

係り受け木に対してスコアを再帰的に定義する．準備として，$\boldsymbol{w} = w_1, \cdots, w_N$ に対して，図 2 に示すように，位置 $0, \cdots, N$ を考える．そして，$\mathrm{score}(i, j, k)$ を位置 i から k までを被覆する木のスコアとする．ここで，j は木の分

割位置を表す．$j < k$ のとき，木が $i, \cdots, j$ と $j, \cdots k$ を被覆する木に分割できる．$j = k$ のときは，それ以上分割できず，また $\mathrm{score}(i, j, k) = 0$ である．$i = 0$ かつ $k = N$ のときは，$\boldsymbol{w}$ 全体を被覆する木を表す．

$j < k$ のとき，$\mathrm{score}(i, j, k)$ は，$\mathrm{score}(i, *, j)$, $\mathrm{score}(j, *, k)$，およびエッジのスコアとスパンのスコアの総和である．エッジのスコア $\mathrm{edgeScore}(j, k)$ は，w_j と w_k の間のエッジにスコアを与える．一方，スパンのスコア $\mathrm{spanScore}(i, k)$ はスパン $w_{i+1}, \cdots, w_k$ にスコアを与える．例として，図 2 の名詞句に対するスコアを以下に示す．

$$\begin{aligned}
\mathrm{score}(0,2,3) &= \mathrm{score}(0,1,2) + \mathrm{score}(2,3,3) \\
&\quad + \mathrm{edgeScore}(2,3) \\
&\quad + \mathrm{spanScore}(0,3) \\
\mathrm{score}(0,1,2) &= \mathrm{score}(0,1,1) + \mathrm{score}(1,2,2) \\
&\quad + \mathrm{edgeScore}(1,2) \\
&\quad + \mathrm{spanScore}(0,2) \\
\mathrm{score}(0,1,1) &= \mathrm{score}(1,2,2) = \mathrm{score}(2,3,3) = 0
\end{aligned}$$

edgeScore と spanScore は特徴量ベクトルと重みベクトルの内積により定義される．重みベクトルは，訓練データが与えられたとき，Passive-Aggressive アルゴリズム [4] を用いて求められる．

図 3 に特徴量を示す．これは [21] で用いられた特徴量を一部変更したものである．最右列のみがスパンの特徴量で，残りはエッジの特徴量である．$\langle * \rangle$ は複数の特徴量に展開されるテンプレートを表す．コロンの左辺は特徴量の名前，右辺はその値を表す．右辺が省略された場合はバイナリ特徴量である．l_i は w_i を小文字で正規化した表記，p_i は w_i の品詞タグを表す．$t = k - j$ は w_j と w_k の間の距離（1, 2, 3, 4 or ≥ 5）を表す．$s = k - i + 1$ はスパンの幅（2, 3, 4, 5 or ≥ 6）を表す．$\mathrm{log1p}(x) = \log(1 + x)$ であり，$x \geq 1$ に対して正の値を返す．c_* は，大規模タグなしコーパスで計算された頻度を返す．$\mathrm{c}_{\mathrm{TWNC}}(l_j, l_k)$ は l_j, l_k が 2 単語の最長名詞句として出現した回数，$\mathrm{c}_{\mathrm{LTW}}(l_j, l_k)$ は l_j, l_k が最長名詞句の末尾 2 単語として出現した回数，$\mathrm{c}_{\mathrm{SPAN}}(l_{i+1}, \cdots, l_j)$ は $l_{i+1}, \cdots, l_j$ が最長名詞句として出現した回数を返す．

大規模タグなしコーパスとしては，実験では 30 億文からなるウェブコーパスを用いた．このコーパスは [13] に示された手法で自動編纂されたものである．このコーパスから，3.2 節および 3.3 節に示した手法で最長名詞句を抽出し，そこからさらに上記の統計を計算する．

名詞句解析器の性能を確認するために小規模実験を行った．評価には Penn Treebank の WSJ 部分を正解データとして利用した．最初に WSJ に名詞句アノテーションパッチ [31] を適用し，次に LTH converter[*6] を用いて各文を係り受け木に変換した．同時に，3.3 節に述べた手法で，3 単語以上からなる，すなわち構造に曖昧性のある最長名詞句を抽出した．それらの最長名詞句に対して，文の係り受け木から得られる係り受け関係を付与した．ここで，主辞後置性が守られない名詞句を除外した．従来研究と同様に，2–21 部を訓練に，23 部を評価に用いた．スコアを最大とする木の探索には動的計画法を用いた．正解品詞タグつきの正解名詞句の単語列を与えたとき，この名詞句解析器は，ラベルなし係り受けスコア（UAS）で 99.19%を得た．ただし，最後から 2 番目の単語は常に最後の単語に係るので除外すると，98.49%となる．以下で用いる名詞句解析器は，WSJ 全体を用いて訓練した．

[*6] http://nlp.cs.lth.se/software/treebank_converter/

5.2 擬似頻度

この名詞句解析を用いてフレーズベース TF を定義する．単語ベース TF-IDF 法と同じく，最長名詞句は高精度に抽出されていると仮定し，最長名詞句に頻度 1 を与える．同時に，部分名詞句に対しても適当な擬似頻度を与える．この際，名詞句解析器が自然と考える部分名詞句には大きな擬似頻度を，そうでない候補には小さな擬似頻度を与える．

擬似頻度の割り当ては，内側外側アルゴリズムに似た動的計画法によって行う．準備として，スコアの総和 $\mathrm{scoreS}(i, k)$ を再帰的に定義する．このスコアは，部分名詞句 $w_{i+1}, \cdots, w_k$ がどの程度自然なまとまりかを表す．

$$\mathrm{scoreS}(i,k) = \begin{cases} 0 & \text{if } i+1 = k \\ \sum_{j=i+1}^{k} \mathrm{scoreE}(i,j,k) & \text{otherwise} \end{cases}$$

$$\begin{aligned}
\mathrm{scoreE}(i,j,k) &= \mathrm{scoreS}(i,j) + \mathrm{scoreS}(j,k) \\
&\quad + \mathrm{edgeScore}(j,k) \\
&\quad + \mathrm{spanScore}(i,k)
\end{aligned}$$

このスコアは下から上に求めていく．

次に，今度は上から下に擬似頻度を分配する．$f_{i,k}$ $(0 < f_{i,k} \leq 1)$ を $w_{i+1}, \cdots, w_k$ に対する擬似頻度とする．ただし，最長名詞句の擬似頻度は 1 とする（$f_{0,N} = 1$）．$f_{i,k}$ は，$\mathrm{scoreE}(i, j, k)$ に基づき，まず一時変数 $g_{i,j,k}$ $(i < j < k)$ に分配される．

$$g_{i,j,k} \leftarrow f_{i,k} \times d \times \frac{exp(\mathrm{scoreE}(i,j,k))}{\sum_j exp(\mathrm{scoreE}(i,j,k))},$$

ここで，d はあらかじめ定義された割引係数とする $(0 \leq d \leq 1)$．続いて，各 $g_{i,j,k}$ は $f_{i,j}$ および $f_{j,k}$ に足しあわされる．

$$\begin{aligned}
f_{i,j} &\leftarrow f_{i,j} + g_{i,j,k} \\
f_{j,k} &\leftarrow f_{j,k} + g_{i,j,k}
\end{aligned}$$

$\langle t \rangle$	$\langle l_j, l_k \rangle$	TWNC : $\log 1\mathrm{p}(\mathrm{c}_{\mathrm{TWNC}}(l_j, l_k))$	$\langle l_{i+1}, \cdots, l_k \rangle$
$\langle l_j, t \rangle$	$\langle l_j, l_k, t \rangle$	LTW : $\log 1\mathrm{p}(\mathrm{c}_{\mathrm{LTW}}(l_j, l_k))$	$\langle p_{i+1}, \cdots, p_k \rangle$
$\langle l_k, t \rangle$	$\langle p_j, p_k, t \rangle$		$\langle s \rangle$: $\log 1\mathrm{p}(\mathrm{c}_{\mathrm{SPAN}}(l_{i+1}, \cdots, l_k))$

図 3: score(i, j, k) に対する特徴量
Fig. 3 Features for score(i, j, k).

このようにして，$d > 0$ のとき，すべての部分名詞句に対して 0 以上の擬似頻度が与えられる．ただし，最後の単語が名詞でない場合は改めて頻度 0 とし，出力から除外する．擬似頻度の分配には softmax 関数を用いている．これにより，名詞句分類器が自然と考える部分列により大きな擬似頻度が分配される．つまり，擬似頻度は文法性を反映している．割引係数 d は，どの程度最長名詞句を部分名詞句より優先するかを制御する．$d = 0$ のときは最長名詞句のみを考慮する．

5.3 フレーズベース TF

擬似頻度をもとにフレーズベース TF を定義する．あるキーフレーズ候補たる単語列 $\boldsymbol{w}$ に着目したとき，文書中のその出現を収集する．$f_1, f_2, \cdots, f_T$ を doc における $\boldsymbol{w}$ の各出現の擬似頻度としたとき，フレーズベース TF はそれらの総和として定義される．

$$\mathrm{phraseTF}_{doc}(\boldsymbol{w}) = \sum_{i=1}^{T} f_i$$

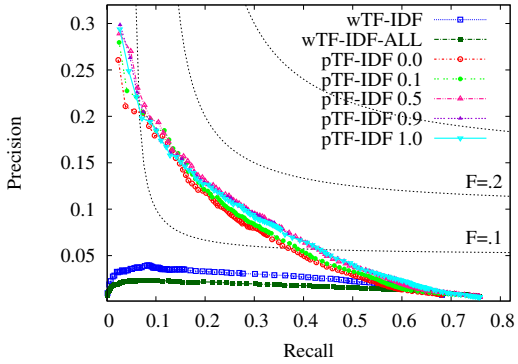
すでに見たように，文法的でないキーフレーズ候補に対しては小さな擬似頻度が与えられるため，その総和も小さくなると期待される．それに加えて，構文的名詞句に対しても小さな値を与えることが期待される．なぜなら，構文的名詞句は，その場で形成されるため，何度も出現しない傾向があるからである．それに対して，重要な語彙的名詞句は繰り返し出現する．

フレーズベース TF は，単語 TF と同様に，一般的なフレーズに対して大きなスコアを与える．一般的なフレーズはキーフレーズとして相応しくないため，単語の場合と同様に，IDF による補正が必要となる．

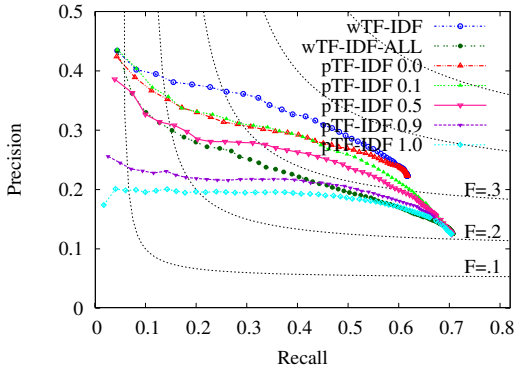
5.4 フレーズベース IDF

フレーズベース IDF phraseIDF$(\boldsymbol{w})$ は，単純に，フレーズが出現する文書数を数えることで得られる．ただし，割引係数 d により振る舞いを変える．$d > 0$ のとき，部分文字列として出現した場合も考慮する．一方，$d = 0$ のとき，最長名詞句として出現した場合のみを考慮する．

フレーズベース IDF は，単語の場合と同様に，一般的なフレーズに小さな値を与え，TF の問題を補正する．単語にない問題，文法性と語彙性について考えると，非文法的なフレーズは，文法的なフレーズとくらべても，ある程度大きな IDF になることが予想される．また，構文的な候補



(a) NUS.



(b) Inspec.

図 4: フレーズベース TF-IDF と単語ベース TF-IDF の比較
Fig. 4 Comparison between phrase-and word-based TF-IDF.

もあまり出現しないことから，比較的大きな IDF になることが予想される．いずれも好ましくない性質だが，現在のところは，フレーズベース TF の補正を期待して，フレーズベース IDF では特に対策を行わない．

単語 TF-IDF と同様に，フレーズベース TF-IDF も TF と IDF の積として定義される．

$$\mathrm{phraseTFIDF}_{doc}(\boldsymbol{w}) = \mathrm{phraseTF}_{doc}(\boldsymbol{w}) \times \mathrm{phraseIDF}(\boldsymbol{w})$$

5.5 結果と議論

図 4 にフレーズベース TF-IDF（pTF-IDF）と単語ベース TF-IDF （wTF-IDF）の比較結果を示す．ここで，フレーズベース TF-IDF の値は割引系数 d を表す．2 個のデータセットで対照的な結果を得た．

NUS では，フレーズベース TF-IDF が単語ベース TF-IDF を大幅に上回る性能を示した．d の値によるフレーズ

ベース TF-IDF 同士の比較では，$d = 0$ が精度がほぼ一貫して最悪となった一方，$d = 0.5$ あるいは $d = 0.9$ がほぼ同程度に良い精度をもたらした．これは，名詞句解析による部分名詞句の活用が精度に貢献していることを意味する．

一方，Inspec については，フレーズベース TF-IDF が単語ベース TF-IDF に一貫して敗れた．しかも，d の値が小さい，すなわち部分文字列の影響が小さいほど高い精度が得られた．単語ベース TF-IDF-ALL と比較すると，$d = 0.1$ および $d = 0.5$ の場合に，フレーズベース TF-IDF が上回った．Inspec では，シソーラスに統制されていないキーフレーズを正解キーフレーズとして用いたが，統制されていないキーフレーズとして最長名詞句が採用される傾向が見られる．また，Inspec の文書は短いため，構文的名詞句自体の出現が多くない．そのため，部分名詞句にスコアを分け与えても副作用しか得られないとみられる．

6. おわりに

本稿では，複数の単語からなる意味的まとまりを一体として認識することを目的に，フレーズベース TF-IDF を提案し，教師なしキーフレーズ抽出に適用した．文書が長い場合には単語ベース TF-IDF を大幅に上回る性能が得られたが，短い場合には下回った．今後は，文書の長さに関わらず頑健に動作するように改良したい．

本稿を含む多くのキーフレーズ抽出の研究は，各キーフレーズ候補に対して独立にスコアを与えてきた．しかし，キーフレーズ一覧をテキスト要約表現と考えると，キーフレーズ同士の関係を考慮し，冗長性を減らすべきかもしれない．また，フレーズの利用はトピックモデルでも盛んに行われており，こちらへの応用も考えている．

謝辞 本研究は一部 JST CREST の支援を受けた．

参考文献

[1] Bergsma, S., Pitler, E. and Lin, D.: Creating Robust Supervised Classifiers via Web-Scale N-Gram Data, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 865–874 (2010).

[2] Cohen, S. B., Blei, D. M. and Smith, N. A.: Variational Inference for Adaptor Grammars, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 564–572 (2010).

[3] Collins, M. J.: A New Statistical Parser Based on Bigram Lexical Dependencies, *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pp. 184–191 (1996).

[4] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S. and Singer, Y.: Online Passive-Aggressive Algorithms, *Journal of Machine Learning Research*, Vol. 7, pp. 551–585 (2006).

[5] Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C. and Nevill-Manning, C. G.: Domain-Specific Keyphrase Extraction, *Prceedings of Sixteenth International Joint Conference on Artificial Intelligence*, pp. 668–673 (1999).

[6] Griffiths, T. L., Steyvers, M. and Tenenbaum, J. B.: Topics in semantic representation, *Psychological Review*, Vol. 114, No. 2, pp. 211–244 (2007).

[7] Hasan, K. S. and Ng, V.: Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-Art, *Coling 2010: Posters*, Beijing, China, pp. 365–373 (2010).

[8] Hulth, A.: Improved Automatic Keyword Extraction Given More Linguistic Knowledge, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pp. 216–223 (2003).

[9] Johansson, R. and Nugues, P.: Extended Constituent-to-Dependency Conversion for English, *NODALIDA 2007 Conference Proceedings*, pp. 105–112 (2007).

[10] Johnson, M.: PCFGs, Topic Models, Adaptor Grammars and Learning Topical Collocations and the Structure of Proper Names, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1148–1157 (2010).

[11] Kageura, K. and Umino, B.: Methods of automatic term recognition: A review, *Terminology*, Vol. 3, No. 2, pp. 259–289 (1996).

[12] Kageura, K., Yoshioka, M., Takeuchi, K., Koyama, T., Tsuji, K., Yoshikane, F. and Okada, M.: Overview of TMREC Tasks, *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, p. 415 (1999).

[13] Kawahara, D. and Kurohashi, S.: Case Frame Compilation from the Web using High-Performance Computing, *Proceedings of The 5th International Conference on Language Resources and Evaluation (LREC-06)*, pp. 1344–1347 (2006).

[14] Kim, S. N., Medelyan, O., Kan, M.-Y. and Baldwin, T.: SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles, *Proceedings of the 5th SIGLEX Workshop on Semantic Evaluation*, pp. 21–26 (2010).

[15] Lapata, M. and Keller, F.: The Web as a Baseline: Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP Tasks, *HLT-NAACL 2004: Main Proceedings*, pp. 121–128 (2004).

[16] Lauer, M.: Corpus Statistics Meet the Noun Compound: Some Empirical Results, *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 47–54 (1995).

[17] Lindsey, R., Headden, W. and Stipicevic, M.: A Phrase-Discovering Topic Model Using Hierarchical Pitman-Yor Processes, *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 214–222 (2012).

[18] Liu, Z., Li, P., Zheng, Y. and Sun, M.: Clustering to Find Exemplar Terms for Keyphrase Extraction, *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pp. 257–266 (2009).

[19] Marcus, M. P., Marcinkiewicz, M. A. and Santorini, B.: Building a Large Annotated Corpus of English: the Penn Treebank, *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330 (1993).

[20] Mihalcea, R. and Tarau, P.: TextRank: Bringing Order into Texts, *Proceedings of EMNLP 2004*, pp. 404–411 (2004).

[21] Murawaki, Y. and Kurohashi, S.: Semi-Supervised Noun Compound Analysis with Edge and Span Features, *Proceedings of COLING 2012*, pp. 1915–1932 (2012).

[22] Nakagawa, H. and Mori, T.: A simple but powerful automatic term extraction method, *COLING-02 on COMPUTERM 2002: Second International Workshop on Computational Terminology - Volume 14*, pp. 29–35 (2002).

[23] Nakov, P. and Hearst, M.: Search Engine Statistics Beyond the n-Gram: Application to Noun Compound Bracketing, *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pp. 17–24 (2005).

[24] Nguyen, T. D. and Kan, M.-Y.: Keyphrase Extraction in Scientific Publications, *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, Lecture Notes in Computer Science, Vol. 4822, Springer Berlin Heidelberg, pp. 317–326 (2007).

[25] Pitler, E., Bergsma, S., Lin, D. and Church, K.: Using Web-scale N-grams to Improve Base NP Parsing Performance, *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 886–894 (2010).

[26] Sang, E. F. T. K. and Buchholz, S.: Introduction to the CoNLL-2000 Shared Task: Chunking, *Proceedings of CoNLL-2000 and LLL-2000*, pp. 127–132 (2000).

[27] Tomokiyo, T. and Hurst, M.: A Language Model Approach to Keyphrase Extraction, *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pp. 33–40 (2003).

[28] Tsatsaronis, G., Varlamis, I. and Nørvåg, K.: SemanticRank: Ranking Keywords and Sentences Using Semantic Graphs, *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 1074–1082 (2010).

[29] Tsuruoka, Y., Miyao, Y. and Kazama, J.: Learning with Lookahead: Can History-Based Models Rival Globally Optimized Models?, *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pp. 238–246 (2011).

[30] Turney, P. D.: Learning Algorithms for Keyphrase Extraction, *Information Retrieval*, Vol. 2, pp. 303–336 (2000).

[31] Vadas, D. and Curran, J.: Adding Noun Phrase Structure to the Penn Treebank, *Proc. of ACL*, pp. 240–247 (2007).

[32] Wan, X. and Xiao, J.: Single document keyphrase extraction using neighborhood knowledge, *Proceedings of the 23rd AAAI Conference on Artificial Intelligence - Volume 2*, pp. 855–860 (2008).

[33] Wang, X., McCallum, A. and Wei, X.: Topical N-Grams: Phrase and Topic Discovery, with an Application to Information Retrieval, *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pp. 697–702 (2007).

[34] Yamada, H. and Matsumoto, Y.: Statistical Dependency Analysis with Support Vector Machines, *Proceedings of the 8th International Workshop on Parsing Technologies*, pp. 195–206 (2003).