2015年度
修士論文

時系列テキストストリームからの
オブジェクト間の関係抽出

神戸大学システム情報学研究科
計算科学専攻

川原 駿

指導教員		上原 邦昭　教授
審査教員	主査	上原 邦昭　教授
	副査	羅 志偉　教授
	副査	中村 匡秀　准教授
	副査	関 和広　准教授

2016年2月8日

Relation Extraction between objects from time series text streams.

Shun Kawahara

Abstract

Nowadays, large knowledge bases, such as Wikipedia, are widely used as a quick reference tool to find all kinds of information in our daily lives. Such knowledge bases can be utilized to improve the performance of various information processing tasks, such as information retrieval. Existing knowledge bases are typically written and maintained by a group of voluntary editors. However, it is practically impossible for the editors to manually monitor numerous web documents. To deal with this issue, we use relation extraction that is the natural language processing technique of automatically extracting useful information from unstructured documents such as news texts. It is expected to automatize updating process of knowledge bases by introducing relation extraction. Existing methods of relation extraction is effective against cumulative web documents. However, we are able to find out information from SNS such as Twitter in realtime. Therefore, it is desired for relation extraction to be carried out in realtime. Furthermore, Existing methods do not discriminate novel information from known one that is unnecessary for updating knowledge bases. This paper proposes effective relation extraction technique against time series web document streams. We incorporate semantic features into state-of-the-art method against cumulative web documents. In addition, our method is judges whether extracted information is novel one by taking advantage of similarity calculation method between documents that is based on vector representation of the word. On a publicly available time series web document data set, the validity of the proposed method is demonstrated.

時系列テキストストリームからの
オブジェクト間の関係抽出

川原 駿

要 旨

あらゆる情報を活用するためのツールとして，Wikipedia に代表されるような知識ベースが幅広く利用されている．その用途は幅広く，人々が百科事典のように利用することはもちろん，情報検索などの自然言語処理の分野でも活用されている．知識ベースの記事は手作業により随時更新されることでその質が保たれているが，ニュース記事などのウェブ文書が日々大量に発信されている昨今では，手作業による更新には限界がある．こうした問題に対応するために，関係抽出と呼ばれる自然言語処理の手法を利用することがあげられる．関係抽出を用いると，ニュース記事などの生のテキストから有用な情報を機械的に抽出することが可能であるため，知識ベースの更新プロセスの自動化が期待できる．従来の関係抽出は大量の文書を利用するバッチ処理である．大量の文書から関係抽出に有用な情報を大量に獲得することで，ウェブ文書に多い崩れた英文などのノイズの影響を小さくしている．一方で，近年では Twitter などの SNS からほぼリアルタイムに情報を得ることが可能となっており，関係抽出も逐次的に行われることが望ましいが，ノイズの処理が問題となる．また，従来の関係抽出の手法は，新たな情報も古い情報も区別せずに抽出を行う．既に知っているような情報は知識ベースの更新には不要であるため，それらは区別されることが望ましい．そこで本研究では，従来の関係抽出において高い性能を示している枠組みを使用しつつ，崩れた英文に対して有効な素性を取り入れることで時系列テキストストリームからの関係抽出を実現する手法を提案する．また，単語のベクトル表現に基づき文章の類似度を計算する手法を応用し，新情報を区別することを試みる．時系列テキストコーパスを用いて実際に関係抽出を行い，提案手法の有効性を評価する．

目 次

表 目 次

図 目 次

第1章　序論

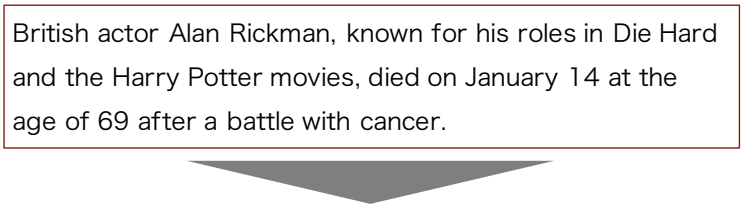
ありとあらゆる情報を参照・活用するためのツールとして，Wikipedia や Freebase に代表されるような知識ベース（knowledge base）が幅広く利用されている．知識ベースとは，事実や常識，経験などの知識をオンライン上でデータベース化したものである．人々が百科事典のように利用することはもちろん，クエリ拡張 [7, 27]，エンティティ・リンキング [17]，質問応答 [6]，情報検索 [9] といった自然言語処理の種々のタスクの性能改善にも用いられている．例えば情報検索の分野では，Google は実際に Knowledge Graph [24] と呼ばれる知識ベースを検索技術に導入している．自然言語処理の分野において，知識ベースの情報の質を維持することは非常に重要とされている．そのためには，古くなった情報を最新のものに更新したり，新たな情報を随時追加していく必要がある．

知識ベースの記事は手作業により更新されることが一般的であり，管理者が新たな情報をその目で確かめて記事を編集するのである．例えば，英語版 Wikipedia の記事の編集者（管理者）はおよそ 1300 人存在しているが [1]，これに対して記事数は 450 万件以上にもなる．単純計算すると 1 人当たり 3500 件近い記事を担当しなければならないことになる．さらに，Yahoo!ニュースや新聞紙各社のウェブサイトなどのインターネットメディアの普及により，ニュース記事などのウェブ文書が日々大量に発信されている．Wikipedia の記事の編集者は，これらの文書を逐一確認して記事に反映させなければならないものの，発信される情報が多すぎるため，現実には多くの記事において情報が反映されるまでに 1 年もの時間を要することも少なくない [11].

こうした問題に対応するために，関係抽出（Relation Extraction）と呼ばれる自然言語処理の手法を利用することがあげられる．関係抽出とは Fig. 1.1 のように生のテキストから有用な情報を構造化テキストとして抽出する手法であ

[1] http://en.wikipedia.org/wiki/List_of_Wikipedias 参照

る．生のテキストに対して固有表現抽出や構文解析を行い，オブジェクト間の



Alan Rickman	
occupation	actor
filmography	Die Hard, Harry Potter
death date	January 14
death cause	cancer

Fig. 1.1: An example of relation extraction.

関係を推定することにより，有用な情報を整理された形で抽出することを可能としている．このように，膨大な数の Web 文書から自動的に情報を抽出することができれば，知識ベースの更新プロセスの一部もしくは全てを自動化することが可能となる．現在提案されている関係抽出の手法は，ある程度の期間蓄積された大量のウェブ文書を利用するバッチ処理型である．ウェブ文書の中には崩れた英文やノイズを多く含むものが多く，関係抽出性能の悪化の原因となる．そこで，大量のウェブ文書を用いることで関係抽出に有用な情報を大量に獲得し，ノイズの影響を相対的に小さくすることで性能の悪化を防いでいる．

近年，Twitter や Facebook などのソーシャル・ネットワーキング・サービス（SNS）が普及したことにより，タイムラインのような時系列テキストストリームを眺めれば，ほぼリアルタイムに情報を得ることが可能となっている．情報検索などにおいても，現在の情報をリアルタイムに反映した検索結果が求められるようになってきているが，そのためには情報検索に活用されている知識ベースの情報をリアルタイムに更新する必要がある．これを実現するには，時系列テキストストリームから逐次的に関係抽出を行わなければならないが，従来手法ではノイズによる関係抽出精度の悪化が問題となる．さらに，従来の関係抽出の手法では新たな情報も古い情報も区別せずに抽出を行う．例えば，Fig. 1.1

は「英国俳優のアラン・リックマンが 2016 年 1 月 14 日に癌で亡くなった」という記事で，2016 年 1 月 14 日に発行されたものである．この記事における新情報は「死没日時（death date）が 1 月 14 日」であることと,「死因（death cause）が癌」であることである.「職業（occupation）が俳優」であることや,「出演映画（filmography）はダイ・ハードとハリー・ポッター」であるという情報は，この記事が発行される以前に知られている情報である．既知の情報は知識ベースの更新には不要であるため，出力を行わないことが望まれるが，従来手法ではこのような古い情報も出力してしまう．

以上を踏まえて本研究では英語を対象として，時系列テキストストリームに対して関係抽出を行う，ストリーム型の関係抽出手法を提案する．バッチ処理型の関係抽出手法として高い性能を示している Distant supervision [19] の枠組みを使用しつつ，使用する素性を工夫することによりストリーム型の関係抽出を実現する．また，Skip-gram モデル [18] による単語のベクトル表現に基づき文章の類似度を計算する手法を応用し，新情報を区別することを試みる．

本稿の構成は以下の通りである．2 章で関係抽出の関連研究として代表的な手法を述べ，ストリーム型の関係抽出に関する国際ワークショップである TREC KBA Stream Slot Filling タスクの詳細を説明する．3 章で提案手法について具体的に説明し，4 章では，時系列テキストストリームに対して提案手法による関係抽出を行う実験を行い，抽出結果について考察する．最後に，5 章で本稿のまとめと今後の課題について述べる．

第2章　関係抽出

情報抽出の中で，ある 2 つのオブジェクトや語句・表現の間の関係を推定する処理を関係抽出という．関係抽出において，オブジェクトや語句・表現のことをエンティティ（entity）と呼ぶ．エンティティには，人物，組織，施設，場所などの固有表現や，名詞句なども含まれる．1 章で示した Fig. 1.1 は，あるニュース記事の生テキストに対して関係抽出の処理を施し，英国俳優のアラン・リックマンと関係のある語句を，その関係とともに抽出したものである．以下では関係抽出の代表的な手法を述べる．

2.1　バッチ型関係抽出の代表的手法

バッチ型の関係抽出を行う手法としては，教師あり学習を用いてエンティティ同士の関係の判別を行う関係抽出器を構築する手法がある．関係抽出器は，エンティティのペアとそれらに関する素性を入力とし，その関係を出力とするような分類器である．入力となる素性としては主に語彙素性（lexical feature）と統語素性（syntactic feature）の 2 つがあり，これらはエンティティペアが共起する文中から獲得する．語彙素性としては，エンティティ間やその前後の単語列とその品詞情報 [25] や，出現パターンを正規表現で表したもの [3] などがある．統語素性としては，係り受け構造中のエンティティ間を結ぶ最短パス [4] や，エンティティペアを含む部分木 [29] などが用いられる．特に，統語素性は関係抽出において有効な素性となることが知られている [19].

学習に使用するデータは人の手によりラベル付けが行われているのでノイズが少なく，高い精度で学習できることが期待できる．しかし，そのためには大量のラベル付きコーパス [1] を用意する必要があるが，人の手によりラベル付け

[1] 自然言語処理の分野において，テキストを大規模に集積したデータセットのことをコーパ

を行うのは非常に高コストである．限られたデータで学習を行うため，構築した関係抽出器は学習データのドメインに依存してしまい，汎用性を欠いてしまうことも少なくない．このように，教師あり学習の手法では，用意できる学習データには限りがあることが欠点となる．

一方で，教師なし学習を用いて関係抽出器を構築する手法も提案されている．Banko ら [1] は大規模なテキストコーパスからエンティティ間の単語列を抜き出し，クラスタリング手法を適用することで関係抽出を行っている．この手法であれば，大量のテキストデータさえ用意できればラベル付けを行わなくても学習が可能であるため，教師あり学習の欠点をカバーしていると言える．しかし，この関係抽出器ではラベルを出力することはできないため，人手でクラスタにラベル付けを行う必要がある．

2.2 学習データの獲得

近年では，学習データへのラベル付けを半自動化し，教師あり学習の手法で関係抽出器を学習する手法が主流となっている．ラベル付けが半自動化されることにより，大量のラベル付き学習データを獲得することが比較的容易となる．本節では，ラベル付き学習データを半自動的に獲得することができる手法である，ブートストラッピング法（Bootstrapping）と Distant supervision について説明する．

2.2.1 ブートストラッピング法

ブートストラッピング法 [23] は人手によるラベル付け作業を軽減するためのフレームワークである．ラベル付けを行いたい関係（例：occupation）に属するエンティティペア（例：(Alan Rickman, actor)）をシードとして与え，ラベルなしコーパスからエンティティペアと共起するパターン（例：Alan Rickman *'s occupation is* actor.）を抽出し，抽出した共起パターンを用いて新たなエンティティペア（例：(Tim Cook, CEO of Apple)）を抽出する，といった手順を反復

ス（corpus）と呼ぶ．

的に繰り返すことで，少数のエンティティペアから大規模な共起パターンを再帰的に獲得する．共起パターンは，2.1 節で説明した語彙素性や統語素性であり，ラベル付きデータとみなすことができるので，獲得した共起パターンを用いて教師あり学習の手法を適用することができる．この時，初期の入力に用いるシードはラベル付きデータであるが，用意するのは少数で構わないため，人手による作業コストは低く済ませることができる．

ブートストラッピング法では，反復処理を繰り返していくうちにシードとは関係のないエンティティペアやパターンを抽出してしまう問題が知られており，意味ドリフト（semantic drift）[5] と呼ばれている．意味ドリフトは，ブートストラッピング法の反復過程において，複数の関係に属しうるようなパターンを抽出してしまうことに起因する現象である．例えば，関係 occupation のシードとして (Alan Rickman, actor) を入力すると，反復を繰り返していくうちに「X is Y」というパターンを獲得したとする．このパターンは「Alan Rickman is actor.」のように職業を表す一方で,「Michael Leitch is Japanese.」のように国籍を表すパターンにもなり得るため，エンティティペアとして (Michael Leitch, Japanese) を誤って獲得してしまう．さらに，このペアから「Michael Leitch became Japanese citizen.」という文を抽出し，職業と全く関係のない「X became Y citizen」というパターンを獲得してしまう．一旦関係のないパターンを抽出してしまうと，以後の反復でも同様に無関係のパターンが抽出されてしまう．

2.2.2 Distant supervision

Distant supervision は，用意が簡単なラベル付きデータの情報を手がかりに，全く別のラベル無しデータにラベルを付与する手法であり，Mintz ら [19] により確立された．ラベル付きデータは，エンティティペアとその関係をラベルとした 3 つ組 $(entity_1, entity_2, relation)$ の形式で用意する．そして，ラベル無しコーパスを参照し，エンティティペアが共起する文があればそのペアの関係をラベルとして付与する．このようにして，人手による作業をほぼ介することなく，大量のラベル付き学習データを獲得することができる．

Mintz らの手法では，3 つ組は Freebase から自動的に抽出することにより 180

万以上獲得している．また，ラベル無しコーパスとしてはWikipediaの120万記事分のダンプデータ[2]を用いている．FreebaseやWikipediaは分野に依存しないテキストデータであるため，構築した関係抽出器が学習データのドメインに依存することは起こりにくいと考えられる．またそれ以外の任意のテキストデータを使用することができるため，様々な分野のテキストに対して適用できることもDistant supervisionの利点である．一方で，ラベル付けを機械的に行うため，その精度はどうしても人手で行った場合に比べて低くなり，多くのノイズが介入するという欠点もある．しかし，大規模なラベル無しコーパスを用いることで素性の表現がより豊富となるので，結果として人手でラベル付けした小規模な学習データによる教師あり学習よりも性能を向上させることが可能となっている．また，ブートストラッピング法のような繰り返し処理を行うこともないため，意味ドリフトの問題も生じない，

Distant supervisionにより獲得したラベル付きデータにより学習した関係抽出器は，関係抽出のタスクにおいて実際に高い性能を示している．NISTが主催するTAC (Text Analysis Conference) 2014のトラックの1つであるKBP (Knowledge Base Population) のESF (English Slot Filling) タスク [26] においては，参加した18チーム中14チームがDistant supervisionを使用し，性能が高かった上位3チームのシステムは全てDistant supervisionを用いたものであった．本研究においても，Distant supervisionの枠組みを利用したストリーム型の関係抽出手法を提案する．

2.3 TREC KBA Stream Slot Fillingタスク

時系列テキストストリームを対象とした関係抽出を行うタスクとして，Stream Slot Filling (SSF) がある．これはNISTが主催するTREC (Text REtrieval Conference) のKnowledge Base Acceleration (KBA) トラックにおいて，2013年 [10] 及び2014年 [12] にサブタスクとして設定されたものである．KBAトラックは，テキストストリームを監視して知識ベースを更新すべき情報を検出したら推薦を行うことにより，人手による知識ベースの管理効率を向上させることを共通

[2]句読点を含む総単語数は601,600,703単語.

の目標としている．SSF タスクでは，予めターゲットエンティティとして特定の人物・組織・施設などが設定されている．さらに，その各々に対してスロットと呼ばれる抽出対象の関係が与えられている．英語の Web 文書からなる大規模な時系列コーパスから，スロット値となり得るようなフレーズを関係抽出により獲得する．さらに 2013 年のタスクでは，抽出したフレーズが新しいものであるかどうかを判定すること（新情報判別）が求められる．本研究では，2013 年のタスクの設定に基づいて評価を行う．

2013 年のタスクでは，エンティティ間の単語列をクエリに用いた全文検索エンジン Indri による曖昧検索 [20]，推論ルール（例：X の創設者が Y ならば，Y による創設企業は X である）を用いた手法 [21]，係り受けパスを素性としたロジスティック回帰分類器 [13]，係り受けパスにトリガー語と呼ばれる対象の関係をよく表した単語（例：death cause という関係のトリガー語は ‘died’）を導入した手法 [28]，ブートストラッピング法 [13] などが提案された．しかし，上記のいずれの手法も性能は悪く，原因として崩れた英文などのノイズを多く含むウェブ文書に対しては，従来の関係抽出で有効とされた手法がうまく機能しなかったことが考えられる [10]．また，新情報判別においては，過去に抽出されたフレーズと完全一致するかどうか [21] や，編集距離（edit distance） [13] といった文字列ベースの手法のみしか利用されておらず，こちらも低い性能であった．

本研究では，上記のような特徴をもつ Web 文書に対しても有効な素性を用いた関係抽出器を提案する．また，Skip-gram モデル [18] による単語のベクトル表現を導入し，フレーズの意味情報を考慮した新情報判別手法を提案する．

第3章　提案手法

本章では，時系列テキストストリームに対して関係抽出を行う，ストリーム型の関係抽出手法について述べる．本手法は大きく以下の 3 フェーズに分けることができる．

フェーズ 1. 関係抽出器の構築

フェーズ 2. 時系列テキストストリームに対する関係抽出

フェーズ 3. 新情報の判別

フェーズ 1 では，Distant supervision により獲得したラベル付きデータから素性抽出を行い，2 つの関係抽出器を構築する．フェーズ 2 では，フェーズ 1 で学習した関係抽出器を用いてテキストストリームから関係抽出を行う．最後にフェーズ 3 で，抽出した関係が新情報であるかを判別した後，最終的な出力を得る．以下では，各フェーズの詳細について説明する．

3.1　関係抽出器の構築

本フェーズの概要を Fig. 3.1 に示す．まず，Distant supervision に必要なラベル付きデータ（labeled entity pairs）を Wikipedia の Infobox から抽出する．抽出したラベル付きデータと Wikipedia のダンプデータによる Distant supervision を行い，ラベル付き学習データ（labeled sentences）を大量に獲得する．続いて，語彙・統語素性による抽出器（relation classifier by lex/syn）と，単語の意味情報を素性とした抽出器（relation classifier by words）を構築する．構築した 2 つの分類器による予測を組み合わせることにより，最終的な関係の予測を行う．以下でその詳細について説明する．

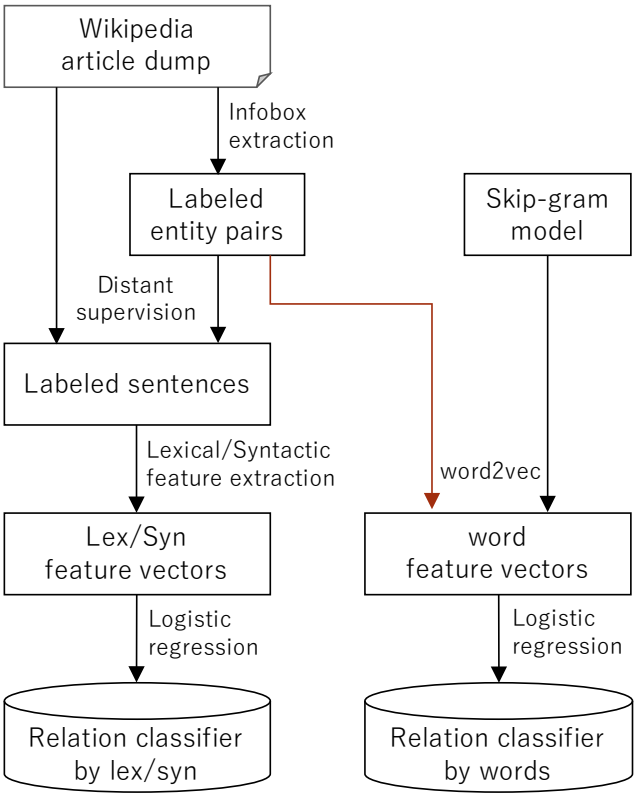


Fig. 3.1: Overview of learning phase of relation classifiers.

3.1.1 Distant supervision の適用

Distant supervision を用いるには 2.2.2 節で述べたように，大規模ラベル無しコーパスと，エンティティペアとその関係をラベルとした 3 つ組の 2 つが必要である．前者には，Mintz ら [19] の手法と同様に Wikipedia の記事のダンプデータを使用する．後者には，Wikipedia の記事中に存在する Infobox と呼ばれる要約欄のデータを使用した．Infobox には関連する記事に共通する要素が属性（attribute）と値（value）を組とした構造化テキストの形で整理されている．これを利用すれば，以下のように 3 つ組を獲得することができる．

$$(entity_1, entity_2, relation) = (article\_title, value, attribute) \tag{3.1}$$

このようにして得られた3つ組とWikipediaのダンプデータを用いて，Distant supervisionを適用することにより，大量のラベル付きの文を獲得する．

なお，Infoboxから3つ組を獲得する処理，Distant supervisionのいずれにおいても，大規模コーパスを扱う必要がある．そこで，処理を高速化するためにApache Hadoop [2] による並列分散処理を利用している．

3.1.2 語彙・統語素性による抽出器

語彙・統語素性による抽出器は，関係が未知のエンティティペアが共起する文から抽出した語彙素性・統語素性を入力すると，その関係を予測した結果を出力する多クラス分類器である．

素性抽出 本分類器の素性としては，以下の語彙情報と統語情報を使用する．これらはMintzら [19] が用いたものと同様である．

1. エンティティ間の単語列とその品詞（語彙素性）
2. 1. + 前後1単語を含めたもの
3. 1. + 前後2単語を含めたもの
4. 係り受け構造中のエンティティ間を結ぶ最短パス（統語素性）
5. 4. + エンティティとのパスが存在する全てのノード

品詞推定や係り受け解析には，フリーの自然言語処理ツールであるStanford CoreNLP [15] を使用する．品詞はPenn Treebankで定義されたものを付与している[1]．実際に素性抽出と係り受け解析を行った例をそれぞれFig. 3.2及びFig. 3.3に示す．Fig. 3.2中の‘type’は前述した素性の番号と対応している．例のように，エンティティはそのタイプ[2]に変換し，素性の一部として使用する．

[1] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[2] PERSON, LOCATION, ORGANIZATION, DATE, TIME, DURATION, SET, MONEY, NUMBER, ORDINAL, PERCENTのいずれかが付与される．いずれにも当てはまらない場合はOBJECTが付与される．

この素性抽出処理は，Distant supervision により得られた全てのラベル付きの文に対して行われる．

British actor ***Alan Rickman*** died of ***cancer*** yesterday.

type	feature
1	***PERSON*** died/VBD of/IN ***OBJECT***
2	actor/NN ***PERSON*** died/VBD of/IN ***OBJECT*** yesterday/NN
3	British/JJ actor/NN ***PERSON*** died/VBD of/IN ***OBJECT*** yesterday/NN
4	***PERSON*** $\uparrow_{\text{nsubj}}$ died $\downarrow_{\text{nmod}}$ ***OBJECT***
5	British $\uparrow_{\text{amod}}$ ***PERSON*** $\uparrow_{\text{nsubj}}$ died $\downarrow_{\text{nmod}}$ ***OBJECT***
5	actor $\uparrow_{\text{compound}}$ ***PERSON*** $\uparrow_{\text{nsubj}}$ died $\downarrow_{\text{nmod}}$ ***OBJECT***
5	***PERSON*** $\uparrow_{\text{nsubj}}$ died $\downarrow_{\text{nmod}}$ ***OBJECT*** $\downarrow_{\text{case}}$ of
5	British $\uparrow_{\text{amod}}$ ***PERSON*** $\uparrow_{\text{nsubj}}$ died $\downarrow_{\text{nmod}}$ ***OBJECT*** $\downarrow_{\text{case}}$ of
5	actor $\uparrow_{\text{compound}}$ ***PERSON*** $\uparrow_{\text{nsubj}}$ died $\downarrow_{\text{nmod}}$ ***OBJECT*** $\downarrow_{\text{case}}$ of

Fig. 3.2: An example of lexical/syntactic features extraction.

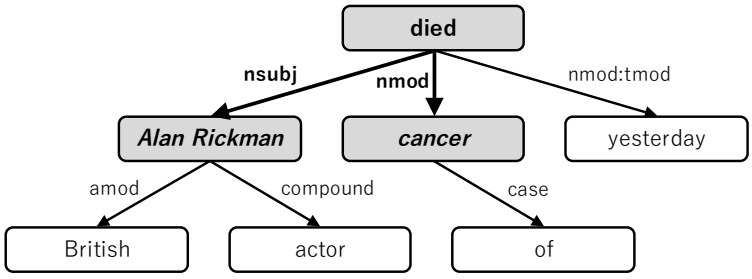


Fig. 3.3: Syntactic dependency parse with shortest dependency path from ‘Alan Rickman’ to ‘cancer’ highlighted in boldface.

素性ベクトルの作成　続いて，素性抽出処理により得られた素性を用いて，ラベル付きの文それぞれの素性ベクトルを作成する．文 S に対する素性ベクトル

は式 (3.2) のように定式化される．

$$\boldsymbol{x_S} = (x_1, x_2, x_3, ..., x_D) \tag{3.2}$$

作成するベクトルの各次元 x_i は 1 つの素性に対応しており，その素性が文から抽出されれば 1，されなければ 0 が素性値となる．例えば，x_3 は語彙素性 'PERSON died/VBD of/IN OBJECT' の有無であるとすると，Fig. 3.2 の文の素性ベクトルの x_3 の値は 1 となる．また，素性ベクトルの次元数 D は，素性抽出処理により全てのラベル付きの文から抽出された相異なる素性の総数となる．

分類器の学習　上記で作成した素性ベクトルを訓練事例として分類器を学習する．本手法では，分類器としてロジスティック回帰（logistic regression） [8] を用いる．予測すべき関係は通常 3 つ以上存在するので，one-vs-the-rest な多クラス分類器として学習する．

ロジスティック回帰により構築される分類器は，式 (3.3) のように，文の素性ベクトル $\boldsymbol{x}$ を入力するとその関係が y である確率を返す．

$$P(y|\boldsymbol{x}) = \frac{1/(1 + e^{-\boldsymbol{w}_y^T \boldsymbol{x}})}{\sum_{m=1}^{k} 1/(1 + e^{-\boldsymbol{w}_m^T \boldsymbol{x}})} \tag{3.3}$$

ここで，k はこの分類器により予測可能な関係の数を表す．また，$\boldsymbol{w} \in \mathbb{R}^D$ は素性に対する重みであり，学習により最適化される．全ての関係に対して式 (3.3) を計算し，最も確率の高い y がこの分類器による関係の予測結果となる．なお，ロジスティック回帰の実装にはフリーの線形分類器実装である LIBLINEAR [8] を使用する．

3.1.3　単語の意味情報を素性とした抽出器

エンティティを表すフレーズを構成する単語の中には，その関係を特徴づけるものがある．例えばあるエンティティが '〜 awards' や '〜 prize' のようなフレーズで表現される場合，もう片方のエンティティ（人物や組織など）が獲得した賞であることが直感的に言える．これは，'awards' や 'prize' が「賞」という概念を持っているからである．あるいは 'actor' や 'writer' というフレーズは

多くの場合人物の職業を表す．これも同様に，‘actor’ や ‘writer’ が「職業」という概念を持っているからである．このように，フレーズを構成する単語の概念からその関係を推定することができると考えられる．そこで，単語の意味や概念を固定長のベクトルで表現することができると言われている，Skip-gram モデルによる単語のベクトル表現 [18] を素性とした関係抽出器を学習することを考える．

Skip-gram モデル　Skip-gram モデルは Fig. 3.4 に示すような構造を持っており，文書中における注目単語 $w(t)$ の前後 c 個の周辺単語 $w(t-c), ..., w(t+c)$ を予測するモデルである．文書集合をもとに周辺単語を正しく予測できるようにベクトルを学習する．つまり，単語列 $w_1, \ldots, w_T$ が与えられた時，Skip-gram の目的関数は以下の対数尤度を最大化することとなる．

$$\frac{1}{W}\sum_{t=1}^{W}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{3.4}$$

そして，以下の softmax 関数を利用して周辺単語の予測を行う．

$$p(w_{t+j}|w_t) = \frac{\exp\left({v'_{w_{t+j}}}^{\mathrm{T}} v_{w_t}\right)}{\sum_{w=1}^{W}\exp({v'_w}^{\mathrm{T}} v_{w_t})} \tag{3.5}$$

式 (3.5) における v_{w_t} が，最終的に獲得したい注目単語 w_t のベクトル表現である．また，式 (3.5) は語彙数 W の大きさに比例して計算量が増大するため，実際には近似手法の階層的 softmax [18] が用いられている．上記モデルから生成された単語のベクトル空間においては，意味的に似た単語ほどベクトル的に近くへ配置されるという性質があることがわかっている．さらに，ベクトル同士の加減算ができるという性質も存在し，例えば，‘king - man + woman = queen’ が成り立つ．この例では王という概念から性別の概念を足し引きすることで女王の概念ベクトルを得ることができている．このように，Skip-gram モデルの学習により生成された単語のベクトル表現は，単語の意味や概念情報を表現していると考えられる．

分類器の学習　単語の意味情報を素性とした抽出器は，エンティティを表すフレーズに含まれる単語のベクトル表現を入力すると，その関係を予測した結果を

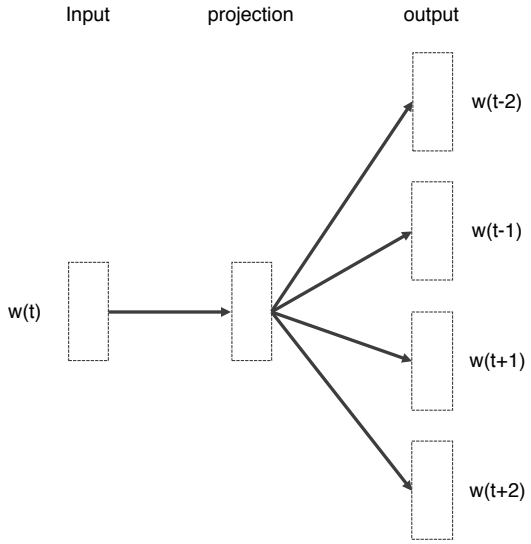


Fig. 3.4: Architecture of the Skip-gram Model.

出力する多クラスのロジスティック回帰分類器として実装する．訓練事例は，エンティティを表すフレーズに含まれる各単語毎のベクトル表現であり，1 単語が 1 事例に対応する．これを 3.1.1 節で得られた 3 つ組の $value$ と $attribute(relation)$ から獲得する．このようにして得られた訓練事例を用いて，語彙・統語素性による抽出器と同様に式 (3.3) の分類器を学習する．

関係の予測　上記で学習した分類器は，ある単語がどの関係を表している可能性が最も高いかを予測する分類器である．実際に関係を予測するエンティティは複数の単語からなるフレーズ $\boldsymbol{W} = (\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}, ...)$ で構成されているため，このままではエンティティ間の関係を予測することはできない．そこで，次式によりフレーズからエンティティ間の関係が y となる確率を計算する．

$$P_{phrase}(y|\boldsymbol{W}) = \frac{\sum_{\boldsymbol{w} \in \boldsymbol{W_y}} P(y|\boldsymbol{w})}{|\boldsymbol{W}|} \tag{3.6}$$

ここで，$|\boldsymbol{W}|$ はフレーズ $\boldsymbol{W}$ を構成する単語の数，$\boldsymbol{W_y} \in \boldsymbol{W}$ は式 (3.3) により関係が y であると予測された単語の集合，$P(y|\boldsymbol{w})$ は式 (3.3) と同様である．式 (3.6) による関係の予測には英文の情報が一切利用されておらず，単語が持つ

意味や概念のみを用いている．このため，Web 文書に多い崩れた英文などのノイズの影響を受けない関係予測を行えることが期待できる．

3.1.4 抽象的関係の導入

3.1.2 節及び 3.1.3 節で述べた 2 つの分類器は，学習を行った関係についてしか予測することはできない．そのため，本来は学習した分類器で予測不可能な関係であるにも関わらず，予測可能な関係から最も可能性の高そうな関係を予測結果として返してしまうことが問題となる．例えば，‘Alan is British.’ という文において，‘Alan’ と ‘British’ の関係を 3.1.2 節の分類器により予測することを考える．この文からは ‘$PERSON \uparrow_{nsubj} OBJECT$’ という素性を獲得することができる．学習済みの分類器によれば，この素性は関係「職業」に対して大きな重みが付いている．すると，分類器は ‘Alan’ と ‘British’ の関係を職業であると予測してしまう．

予測対象外の関係が検出された場合は，予測不可能であるという結果を返すことが望ましい．そこで，抽象的関係 (abstract relation) を導入することを考える．抽象的関係とは,「〜と関係のある人物」や「〜と関係のある日付」のように，エンティティ同士が何らかの関係にあることを表すような関係である．その関係が具体的にどういったものかは考慮しない．対象の関係に加えて抽象的関係も予測できるように分類器を学習する．これにより，上記の例の素性 ‘$PERSON \uparrow_{nsubj} OBJECT$’ のように，PERSON と OBJECT が何らかの関係にあることを示すのみで，具体的にそれが職業であると断定できないような場合に，抽象的関係として予測されることを期待する．抽象的関係が予測結果となった場合を予測不可能とみなすことにより，予測の誤りを減らすことができる．本手法においては，抽象的関係として ‘RelatedPerson’, ‘RelatedLocation’, ‘RelatedOrganization’, ‘RelatedDate’ の 4 つを使用する．

3.2 時系列テキストストリームに対する関係抽出

本フェーズの概要を Fig. 3.5 に示す．テキストストリームから文書を取得すると，始めに関係抽出の対象となっているエンティティ (Target entity) について述べられている文 (Mention sentences) を取り出す．取り出した各文に対して固有表現抽出 (Named Entity Recognition; NER) と名詞句抽出 (Noun phrase extraction) を行い，関係抽出の候補となるエンティティを表すフレーズ (Candidate phrases) を獲得する．次に，候補フレーズ毎に素性抽出を行い，3.1 節で述べた 2 つの分類器を用いてターゲットエンティティとの関係を予測し，結果を候補関係 (Candidate relations) として保持する．候補関係が新情報であるかを判別し，新情報であれば最終的な結果として出力する．以下ではその詳細について説明する．

Mention sentences の抽出　テキストストリームから取得した文書は複数の文から構成されているので，その中から対象となっているエンティティについて述べられている文のみを抽出する．対象のエンティティを示す表記が含まれている文を Mention sentence とみなす．対象のエンティティを示す表記は Wikipedia のリダイレクト情報から取得する．

リダイレクト情報とは，ある記事にアクセスするために記事名で検索を行った時に，その表記が揺れていても正しいページを表示するための情報である．例えば，‘Edson Arantes do Nascimento’ と ‘Pelé’ はそれぞれ本名と愛称であるので同一人物を表すが，何の手がかりもなしには両者が同一であるとコンピュータが判断することはできない．そこで Wikipedia では，「‘Edson Arantes do Nascimento’ というページにアクセスがあったら ‘Pelé’ にリダイレクトする」という情報を内部に持たせておくことで，この問題を解消している．この情報を利用することで，1 つのエンティティに対して複数の表記を取得することができ，表記揺れにも対応することができる．

候補フレーズの抽出　続いて，取り出した各文から関係抽出の候補となるエンティティを表す候補フレーズを獲得する．文中に出現する全ての固有表現と名詞句が候補フレーズの対象となる．固有表現抽出，名詞句の推定はいずれも Stanford

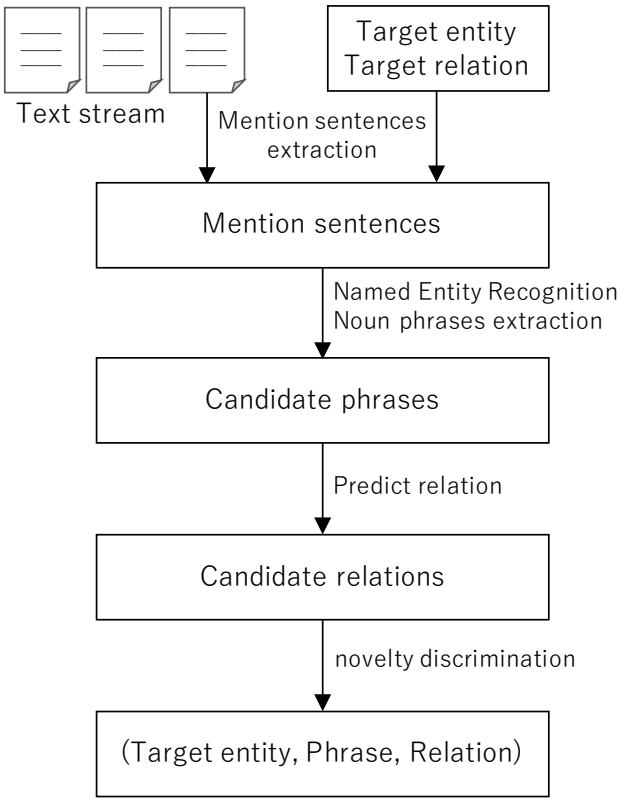


Fig. 3.5: Overview of relation extraction from text streams.

CoreNLP [15] を用いて行う．

関係予測　上記で獲得した候補フレーズ毎に関係予測を行う．候補フレーズ毎に素性抽出を行い，3.1 節で述べた 2 つの分類器を用いてターゲットエンティティとの関係を予測する．2 つの分類器は異なる予測結果を返すため，以下の式に基づき結果を統合する．

$$P_{final}(y|\boldsymbol{x}, \boldsymbol{W}) = \frac{P_{lexsyn}(y|\boldsymbol{x}) + P_{phrase}(y|\boldsymbol{W})}{2} \tag{3.7}$$

$P_{lexsyn}(y|\boldsymbol{x})$ は式 (3.3) により表される語彙・統語素性による分類器によって予測される確率，$P_{phrase}(y|\boldsymbol{W})$ は式 (3.6) により表される単語の意味情報を素性とした分類器によって予測される確率である．各関係ごとに確率を計算し，そ

の値が最大となる関係，すなわち，

$$\arg\max_{y} P_{final}(y|\boldsymbol{x}, \boldsymbol{W}) \tag{3.8}$$

が最終的な予測結果となる．

この関係予測は候補フレーズ毎に行われるため，1 つの関係に複数のフレーズが属することがあり得る．そのような場合は，式 (3.7) による確率が最大のものを，文書から抽出できた関係とみなすことにして，他のフレーズは破棄する．また，その最大の確率を $\max(P_{final})$ とした時，

$$\max(P_{final}) > \phi \tag{3.9}$$

を満たさない場合も．結果を信頼できないものとして破棄する．ただし，ϕ は閾値である．

ここまでの処理により候補関係を得ることができる．次節では，候補関係が新情報であるかを判別する方法について述べる．

3.3 新情報の判別

本研究では，抽出された関係が新情報であるかどうかの判別を「過去に抽出されたフレーズの意味，あるいは抽出元の文脈の同値性判定」に帰着させることを考える．例えば，(Alan Rickman, English actor, occupation) という関係が既知である状況で，Fig. 1.1 に示す関係が新たに抽出されたとする．この時，関係 ‘occupation’ について両者のフレーズを比較すると，ほぼ同じ意味であることが分かる．このように，既知の関係と新たに抽出された関係における 2 つのフレーズが意味的に同値である場合は，抽出された関係も既知である可能性が高いと考えられる．本手法においては，2 つのフレーズを比較した時に，片方のフレーズがもう片方のフレーズに含まれる場合は，意味的に同値であるとみなすことにする．しかし，そうでない場合に意味的に同値でないとは限らない．例えば，‘Edson Arantes do Nascimento’ と ‘Pelé’ は共通単語は存在しないが，全く同じ人物を表している．すなわち意味的に同値である．

そこで，共通単語が存在しない場合は，各フレーズの抽出元となった文脈の同値性により，新情報かどうかを判別することを考える．これは,「同じ文脈に出現する単語やフレーズは同じ意味を持つ」という分布仮説 [16] の考えに基づく．本手法においては，同じ文脈であるかどうかの判定に Word Mover's Distance (WMD) [14] を利用する．

3.3.1 Word Mover ’s Distance

Word Mover ’s Distance (WMD) は，単語のベクトル表現を利用して文の類似度を計算する手法である．単語のベクトル表現を用いることにより，単語間の意味的な近さを考慮した類似度を計算できることが特徴である．

$\mathbf{X} \in \mathbb{R}^{d \times n}$ は各列が単語 i のベクトル表現 $\boldsymbol{x}_i \in \mathbb{R}^d$ を表す行列とする．n は語彙数である．文は normalized bag-of-words (nBOW) $\mathbf{d} \in \mathbb{R}^n$ により表される．nBOW の各次元は単語 i に対応しており，文中に単語 i が c_i 回出現したとすると，$d_i = \frac{c_i}{\sum_{j=1}^{n} c_j}$ で定義される．ただし，ストップワード (付録 A 参照) は除外する．これを利用し，WMD を計算したい 2 文をそれぞれ $\mathbf{d}$ と $\mathbf{d'}$ で表すことにする．また，$\mathbf{T}_{i,j} \geq 0$ を $\mathbf{d}$ 中の単語 i から $\mathbf{d'}$ 中の単語 j への輸送コストとして定義する．さらに，単語 i と単語 j の距離を $c(i,j)$ で表す．距離にはユークリッド距離やコサイン距離が用いられる．この時，nBOW がそれぞれ $\mathbf{d}$ と $\mathbf{d'}$ である 2 文の WMD は次式で定義される．

$$\mathrm{WMD}(\mathbf{d}, \mathbf{d'}) = \frac{\sum_{i,j=1}^{n} \mathbf{T}^*_{ij} c(i,j)}{\sum_{i,j=1}^{n} \mathbf{T}^*_{ij}} \tag{3.10}$$

ただし，$\mathbf{T}^*_{ij}$ は下記の最適化問題により求められる．

$$\begin{aligned} \min_{\mathbf{T} \geq 0} \mathbf{T}^*_{ij} &= \sum_{i,j=1}^{n} \mathbf{T}_{ij} c(i,j) \\ \text{subject to} : \sum_{j=1}^{n} \mathbf{T}_{ij} &= d_i \quad \forall i \in \{1, ..., n\} \\ \sum_{i=1}^{n} \mathbf{T}_{ij} &= d'_j \quad \forall j \in \{1, ..., n\} \end{aligned} \tag{3.11}$$

WMD は距離であるため，値が小さいほど 2 つの文は類似していることになる．

3.3.2 新情報判別のアルゴリズム

テキストストリームから抽出された候補関係は，同じ意味ごとにまとめられて保持される．新たな候補関係が抽出されると，同じ関係として保持されている既知の関係情報との比較が行われる．比較は Fig. 3.6 のチャートに沿って行われ，既知の関係情報のそれぞれに対して，新たな候補関係との類似度 sim が計算される．

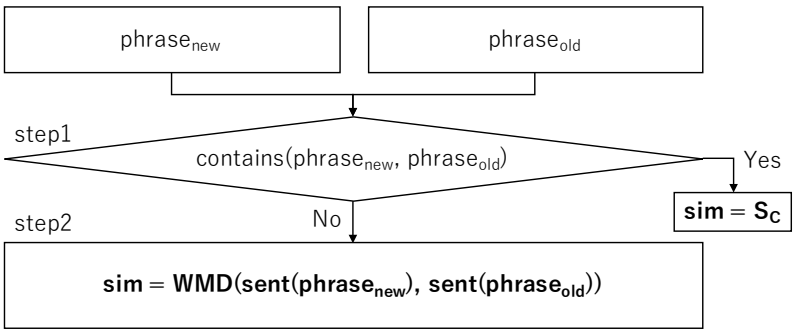


Fig. 3.6: Procedure of novelty discrimination.

step1　2つのフレーズについて，片方のフレーズがもう片方のフレーズに含まれているか (contains) を確認する．含まれている場合は類似度 $sim = S_C$ を与え，含まれていない場合は次のステップを移行する．例えば，‘Alan’ は ‘Alan Rickman’ に含まれているので $sim = S_C$ となるが，‘Alan Sidney Patrick Rickman’ と ‘Alan Rickman’ を比較した場合はいずれのフレーズももう片方には含まれていないので，次のステップへ移行する．

step2　2 つのフレーズの抽出元となった文の WMD を式 (3.10) 及び式 (3.11) により計算し，正負を反転させた値を類似度 sim とする．ただし，2つのフレーズが意味的に同値であるかを判定したいため，計算前にそれらのフレーズを抽出元の文から除外しておく．

新情報であるどうかの決定　上記 step1，2 による類似度は，既知の関係情報それぞれに対して計算される．計算された中で最大の類似度を $\max(sim)$ と定義し，次の式 (3.12) を満たす場合は新たに抽出された関係を新情報であると判別する．

$$\max(sim) < \psi \tag{3.12}$$

式 (3.12) を満たさない場合は既知であると判別し，$\max(sim)$ の値をとったフレーズと同値であるとみなす．ただし，

$$S_C > \psi \tag{3.13}$$

を常に満たしているものとする．また，S_C は step2 で計算されたどの WMD よりも常に大きいものとする．新たに抽出された関係情報はその判別結果とともに保持され，次回以降の新情報判別に利用される．

第4章　評価実験

本研究で提案したストリーム型の関係抽出手法の有効性を検証するため，時系列テキストコーパスを用いて2つの実験を行った．まず，2013年のTREC KBA SSFタスクの設定に基づいた実験を行い，関係抽出の性能を評価した．次に，新情報の判別性能を評価する実験を行った．

以下では，まず実験に使用したデータセットについて説明してから，各実験の詳細を説明し，結果を検証する．

4.1　データセット

本実験では2つのデータセットを使用した．TREC KBA Stream Corpus 2013は時系列テキストコーパスであり，本実験におけるテストデータに用いた．English Wikipediaは英語版Wikipediaの記事のダンプデータであり，学習データとして使用した．

4.1.1　TREC KBA Stream Corpus 2013

TREC KBA Stream Corpus 2013[1]はTREC KBAトラック2013でシステムの評価に使用された時系列テキストコーパスである．2011年10月5日〜2013年2月13日までの約17ヶ月間，1,040,520,595件のウェブ文書で構成されている．それぞれに発行時刻が明示されており，発行時刻順に並べることでテキストストリームデータとして扱うことができる．データセットの内容は，ニュース記事やブログ記事，SNSの投稿などのウェブ上に発行された様々な種類の文書で，言語は英語である．

[1] http://s3.amazonaws.com/aws-publicdatasets/trec/kba/index.html

本研究では，効率的に実験を行うために，コーパスに対して予め前処理を施した．このコーパスには各文書に種々のメタデータが付属しているので，まずはこれらメタデータから提案手法で必要になる，文書の本文，発行時刻，文書識別 ID のみを抽出した．文書の本文はウェブページの HTML ソースコードを抽出したものであるが，予め HTML タグは除去されてある．さらに，英語以外の文書が混入しているため，それらを除去した．また，KBA SSF タスクでは評価用の正解アノテーションが用意されているが，アノテーションは全ての文書に対して行われていない．そのため，アノテーションが行われていない文書も全て除去した．

4.1.2 English Wikipedia

English Wikipedia は，英語版 Wikipedia の全ての記事のダンプデータである．本実験においては 2012 年 1 月 4 日に保存された，およそ 385 万件の記事を下記の目的で使用した．

ラベル付きエンティティペアの抽出 Distant supervision に必要なラベル付きエンティティペアを Infobox の情報から抽出した．この際，SSF タスクで対象となる関係と，Infobox に存在する関係情報を対応付ける必要があり，その結果は付録 B に示しておいた．この対応付け作業は人手により行った．ラベル付きエンティティペアの抽出の結果，Target Relation に関するものを 252,180 件，抽象的関係に関するものを 524,165 件得た．

Distant supervision Distant supervision における大規模ラベル無しコーパスとして使用した．すなわち，実験において使用するラベル付きの文は，全て Wikipedia のダンプデータから取得したものとなる．文の区切りの特定や，日付の正規化には Stanford CoreNLP を用いた．この結果，学習に使用するラベル付きの文として，Target Relation に関するものを 85,539 件，抽象的関係に関するものを 251,333 件得た．

Skip-gram モデルの学習　Skip-gram モデルの学習データとして，予め句読点などのシンボルを除去した本文を使用した．なお，Skip-gram モデルの実装はプログラミング言語 Python のモジュールである gensim [22] の word2vec を用い，パラメータはデフォルトのものを使用した．

エンティティを示す表記の獲得　対象のエンティティについて述べられている文を抽出するために必要な，エンティティの表記をリダイレクト情報から取得した．

4.2　関係抽出の性能評価

本実験は 2013 年の TREC KBA SSF タスクの設定に基づいて行い，提案手法による関係抽出の性能を評価することを目的とする．

4.2.1　評価方法

本実験では，SSF タスクで使用された評価尺度を評価に用いる．以下で SSF タスクにおける評価方法及び評価尺度について説明する．

SSF タスクでは，Stream Corpus を発行時刻順に読み込んで処理を行わなければならない．こうすることで，ウェブ文書の時系列テキストストリームを擬似的に再現する．関係抽出の対象として，21 のエンティティと 4 つの関係'Titles, AwardsWon, CauseOfDeath, DateOfDeath' が与えられる．エンティティ毎に対象となる関係は決まっており，その一覧を付録 C に記しておいた．与えられたエンティティと指定された関係にあるフレーズを抽出し，関係情報（3 つ組）として出力する．

結果を評価する指標としては，全てのターゲットエンティティのマクロ平均による F 値（macro-averaged F1-measure）を用いる．出力結果と評価用の正解アノテーションを比較し，抽出した 3 つ組が正解アノテーションに存在すれば True Positive (TP)，存在しなければ False Positive (FP)，また正解アノテーションに存在する 3 つ組が抽出されなかった場合を False Negative (FN) とす

る．これらの値を各エンティティeごとに求めることでF値を計算することができ，その定義は式 (4.1) で表される．

$$F_1 = \frac{2 \cdot P_{avg} \cdot R_{avg}}{P_{avg} + R_{avg}} \tag{4.1}$$

$$P_{avg} = \frac{1}{|E|} \sum_{e \in E} P(e) \tag{4.2}$$

$$R_{avg} = \frac{1}{|E|} \sum_{e \in E} R(e) \tag{4.3}$$

$$P(e) = \frac{TP(e)}{TP(e) + FP(e)} \tag{4.4}$$

$$R(e) = \frac{TP(e)}{TP(e) + FN(e)} \tag{4.5}$$

ここで，P, Rはそれぞれ適合率と再現率を表し，Eは対象のエンティティの集合である．式 (4.1) の値が高いほど，性能の高い手法であることが言える．

4.2.2 モデルの学習

関係抽出に用いる語彙・統語素性を，Distant supervisionにより得られたラベル付きの文から抽出した結果，その素性ベクトルの次元数は483,070となった．また，ロジスティック回帰で学習を行う際に必要となるパラメータについては，学習データでの5分割交差検定[2]により値を決定した．

学習及びテストにおいて処理する文は，単語数を最大80に制限した．これは，単語数が多い文は崩れた文などのノイズを多く含む場合が多く，このことが原因でStanford CoreNLPによる解析に時間がかかってしまうためである．単語数が80以上の文は，80単語に収まるように文の最初や最後の数単語を除去した．ただし，エンティティペアが除去されないように考慮した．

[2] k-分割交差検定とは，データをkグループに分割し，そのうちの一つを評価用セット，残りの$k-1$グループを学習用セットとして，学習用，評価用セットに割り当てるグループを変えながら，k回の検定を行う評価方法である．

4.2.3 他手法との比較

SSF タスク 2013 に参加していた他チームの手法との比較を行った．その結果を Table. 4.1 に示す．Oracle Baseline [10] は TREC KBA トラックのオーガナイザにより提供されたベースラインである．提案手法と同様に Mention sentence を抽出した後，そのうち最も単語数の多い文から関係を抽出する手法である．PRIS [28] は SSF タスク 2013 において最も性能が高かったシステムである．係り受けパスにトリガー語と呼ばれる対象の関係をよく表した単語を導入した手法を用いている．

提案手法は，2 つの比較手法のいずれよりも高い性能を示した．また，t 検定（両側）を行った結果，提案手法は 2 つの比較手法のいずれとも有意差（有意水準 1%）を確認することができた．このことから，係り受けパスなどの従来の語彙・統語素性に加えて意味素性を加えたことが有効であったことが示唆される．次節以降では，さらなる実験を行うことで意味素性の有効性を検証する．

Table 4.1: Comparison with other systems for relation extraction.

System	Prec.	Rec.	F1
Oracle Baseline	0.047	0.181	0.075
PRIS	0.038	0.007	0.012
Proposed method	**0.253**	**0.313**	**0.280**

4.2.4 各素性の有効性の評価

提案手法で使用した語彙・統語素性（lexical/syntactic），意味素性（semantic）の有効性を評価するために，片方の素性のみを用いた場合と，両方の素性を用いた場合の性能の比較を行った．また，それぞれ抽象的関係（abstract）を用いる場合と用いない場合で比較を行った．その結果を Table. 4.2 及び Table. 4.3 に示す．

Table 4.2: Performance for lexical/syntactic features.

Features	Prec.	Rec.	F1
semantic	0.142	0.212	0.170
+ lexical/syntactic	**0.206**	**0.231**	**0.218**
semantic (with abstract)	0.228	0.273	0.249
+ lexical/syntactic	**0.253**	**0.313**	**0.280**

Table 4.3: Performance for semantic features.

Features	Prec.	Rec.	F1
lexical/syntactic	**0.251**	0.193	**0.218**
+ semantic	0.206	**0.231**	**0.218**
lexical/syntactic (with abstract)	**0.316**	0.156	0.208
+ semantic	0.253	**0.313**	**0.280**

Table. 4.2 では，意味素性のみを用いた場合と，意味素性に加えて語彙・統語素性を用いた場合を比較することにより，語彙・統語素性の有効性を検証した．その結果，抽象的関係の有無にかかわらず約 3〜5 ポイントの F 値の向上が見られた．語彙・統語素性は時系列テキストストリームに対する関係抽出においても有効であると言える．

Table. 4.3 では，語彙・統語素性のみを用いた場合と，語彙・統語素性に加えて意味素性を用いた場合を比較することにより，意味素性の有効性を検証した．その結果，抽象的関係を用いない場合は F 値の向上が見られなかったが，抽象的関係を用いた場合は 7.2 ポイントの F 値の向上が見られた．抽象的関係を用いない場合は再現率が 3.8 ポイント向上したが，適合率は 4.5 ポイントの低下が見られた．このことから，意味素性の追加による抽出できる関係の網羅性を向上させることはできるものの，その精度には問題があると言える．一方で，抽象的関係を用いた場合においても適合率が 6.6 ポイント低下したが，再現率に

おいては 13.9 ポイントもの向上が見られた．意味素性は抽象的関係と組み合わせることにより，非常に有効であることが示唆される．

Fig. 4.1 は，各エンティティ毎に意味素性（+抽象的関係）を加えることにより，Oracle Baseline と比較して F 値がどのように変化したかを示したグラフである．Entity No は付録 C に記されているものと対応している．意味素性の追加により，F 値の差がマイナスやゼロであった 7 つのエンティティのうち，5 つのエンティティ（1, 4, 12, 13, 16）で差がプラスに転じた．一方で，プラスからマイナスに転じたのは 1 つのエンティティのみ（9）であった．このことから，意味素性（+ 抽象的関係）を加える事で，どんなエンティティに対しても性能を発揮する頑健な関係抽出器を構築することができると考えられる．

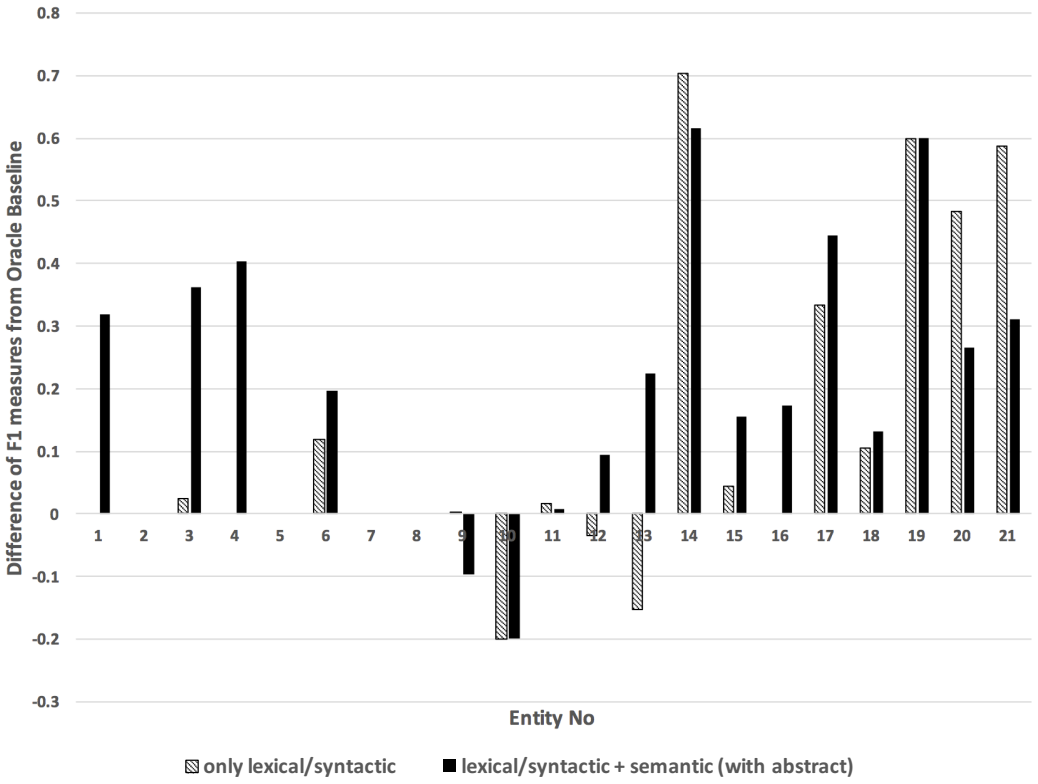


Fig. 4.1: Per-topic difference of F1 measures from Oracle Baseline for each entity.

また，Fig. 4.2 は意味素性を加えることにより，正しく関係抽出を行うことができるようになった例である．斜体で示されている ‘Fernando J. Corbato’ が対象のエンティティ，下線部 ‘2012 Fellow Award honorees’ が抽出すべきフレーズであり，その関係は AwardsWon である．語彙・統語素性のみの場合，このような崩れた英文からは抽出に有効な素性を得ることができなかった．一方で，意味素性であれば「賞」に関係のありそうな ‘Award’ や ‘honorees’ といった単語が有効な素性となり，正しく関係抽出を行うことが可能であった．このように，意味素性は崩れた英文などのノイズを多く含むウェブ文書に対して有効であると考えられる．

Thu Jan 19 , 2012 5:00 am EST MOUNTAIN VIEW, CA, Jan 19 (MARKET WIRE) -- The Computer History Museum (CHM), the world 's leading institution exploring the history of computing and its ongoing impact on society, today announced its **2012 Fellow Award honorees**: Edward A. Feigenbaum, pioneer of artificial intelligence and expert systems; Steve Furber and Sophie Wilson, chief architects of the ARM processor architecture; and ***Fernando J. Corbato***, pioneer of timesharing and the Multics operating system.

Fig. 4.2: An example that semantic features are effective.

4.2.5 抽象的関係の有効性の評価

提案手法で使用した抽象的関係の有効性を評価するために，各素性において抽象的関係を用いた場合と用いなかった場合での性能の比較を行った．その結果を Table. 4.4 に示す．

語彙・統語素性において，抽象的関係を加える事で 1 ポイントの F 値の低下が見られた．ただし，再現率は 3.7 ポイント低下したものの，適合率においては 6.5 ポイントの向上が見られた．これより，抽象的関係を導入した目的であった予測の誤りを減らすという点においては，一定の効果があったことが示唆される．

意味素性においては，抽象的関係の導入により 7.9 ポイントの F 値の向上が見られた．Table. 4.3 の考察においても述べたように，意味素性は抽象的関係と組み合わせることにより，非常に有効であることが言える．

Table 4.4: Performance for abstract relations.

Features	Prec.	Rec.	F1
lexical/syntactic (w/o abstract)	0.251	**0.193**	**0.218**
lexical/syntactic (with abstract)	**0.316**	0.156	0.208
semantic (w/o abstract)	0.142	0.212	0.170
semantic (with abstract)	**0.228**	**0.273**	**0.249**

4.2.6 意味素性の定性的な評価

意味素性による抽出器は，ある単語を入力すると，その単語がどのような関係を表している可能性が最も高いかを予測する分類器を用いている．この分類器に幾つかの単語を入力し，直感的に正しい関係が予測されているかを定性的に評価する実験を行った．具体的には以下の手順で実験を行った．

1. 関係ごとに代表的と思われる単語を筆者の独断で選ぶ．選んだ単語は Table. 4.5 に示す．

2. 選んだ単語の類似語上位 3 件を，word2vec の most_similar 関数により取得する．

3. 上記の単語を分類器に入力した結果が，直感的に正しい関係かどうかを評価する．

実験結果を Fig. 4.3 に示す．このグラフは，各単語ごとにどの関係である可能性が高いか，分類器が出力した確率を棒グラフで表したものである．単語は上から 4 単語ごとに区切ることができ，各区切りの一番上の単語が筆者が選んだ

Table 4.5: Typical words for each relation.

Relation	Word
Titles	actor
AwardsWon	award
CauseOfDeath	cancer
DateOfDeath, RelatedDate	January
RelatedPerson	Alan
RelatedLocation	town
RelatedOrganization	Party

単語，下の 3 つが word2vec により出力された類似語である．このグラフから，ほとんどの単語において直感的に正しいと思われる関係が予測されていることが分かる．また，January, December, July, February は RelatedDate である確率が最も高くなっている．これは，月を表す単語だけでは「死没日時」であるということまでは絞り込めないという直感を反映していると示唆される．ただし，抽象的関係である RelatedDate がなければ，これらの単語だけから「死没日時」が予測されてしまい，誤りが増大していた可能性がある．このように，単語のみから関係を予測する場合は，抽象的関係が大きな役割を果たしていることが分かる．また，直感的に RelatedOrganization に属すると考えられる単語（Party, Progressive, Democratic, Socialist）は，分類器により正しく予測されなかった．しかし，確率が複数の関係に分散しており，これらの単語によって誤分類が起こりうる可能性は比較的低いと考えられる．

4.2.7 関係抽出における閾値

提案手法における関係抽出器は，エンティティ間がある特定の関係となる確率を出力し，その確率と式 (3.9) で表される閾値 ϕ との大小により，その結果を支持するかどうかを決定する．この閾値 ϕ を変化させた時の F 値の変化を Fig. 4.4 に示す．閾値の増加に伴って徐々に F 値が上昇していき，$\phi = 0.22$ の時に F 値

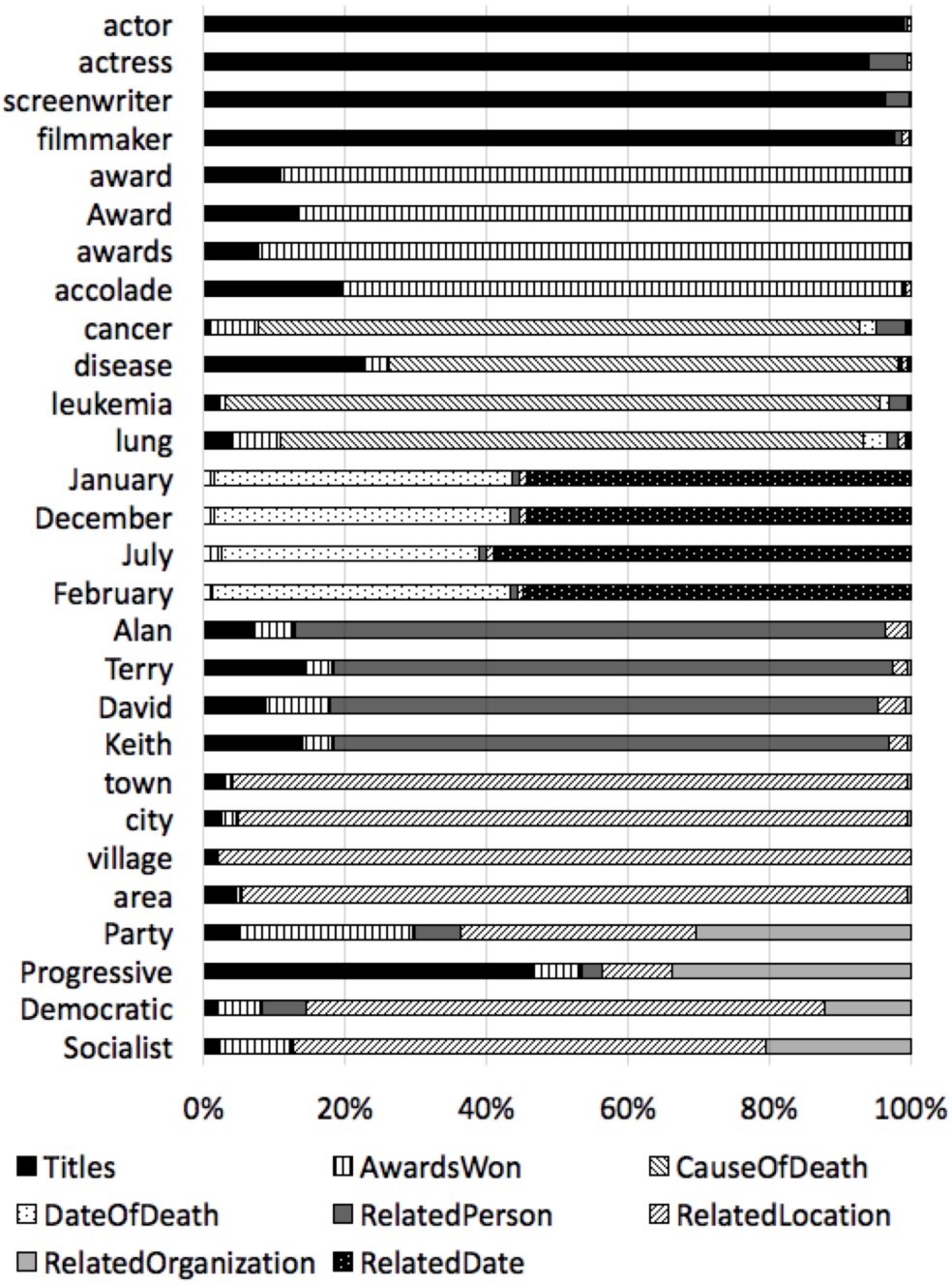


Fig. 4.3: The probabilities for each word.

が最大となった．仮に閾値を設定しなかった（$\phi = 0.0$）場合，最大の F 値と 2.2 ポイントの差があり，適切な閾値を設定することが，提案手法においては重要であることが示唆される．

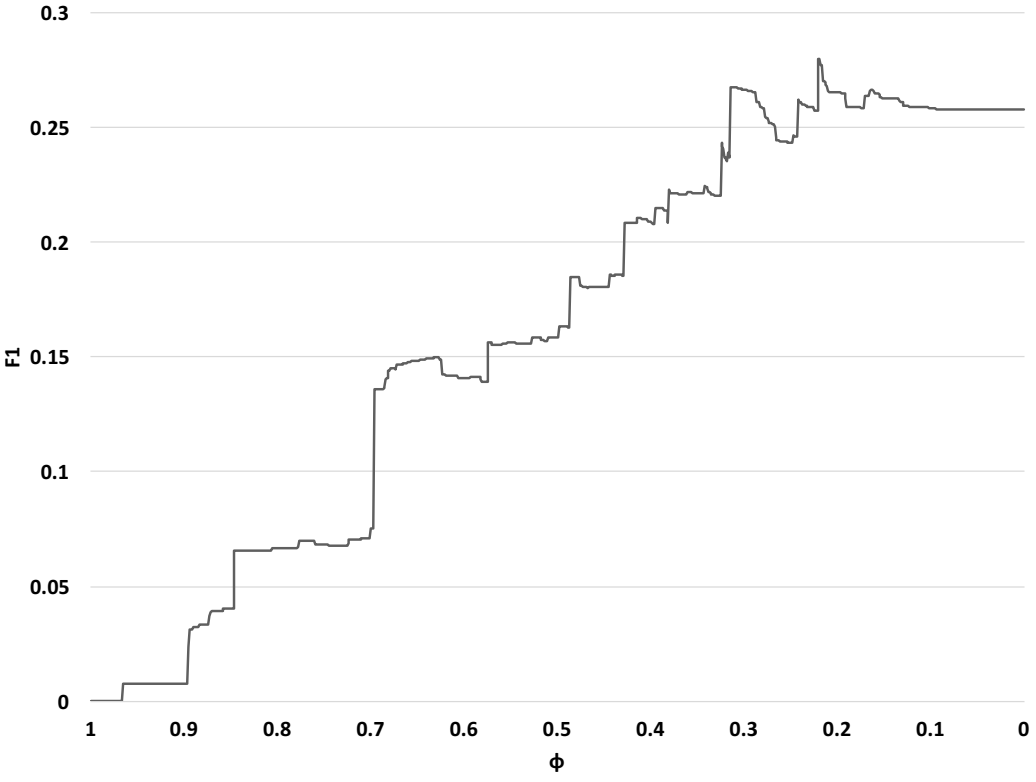


Fig. 4.4: F1 measure vs. threshold ϕ for relation extraction.

4.3 新情報判別の性能評価

本実験では，提案手法による新情報の判別性能を評価することを目的とする．関係抽出の性能評価と同様に，2013 年の TREC KBA SSF タスクの設定に基づいた実験を行った．

4.3.1 評価方法

本実験においても，Stream Corpus を発行時刻順に読み込んで処理を行う．新情報の判別性能の評価のみに焦点を当てるため，関係抽出は正しく行えたものと仮定し，抽出された関係情報が新情報であるかどうかを提案手法により判別する．対象として用いるエンティティと関係は，関係抽出の性能評価で用いたものに加えて，付録 C に記されている全てのエンティティ及び付録 B に記されている全ての関係とする．結果を評価する指標は関係抽出の性能評価と同様に，式 (4.1) で表される全てのターゲットエンティティのマクロ平均による F 値を用いた．

4.3.2 実験結果

実験を行った結果を Table. 4.6 に示す．Baseline は同じ関係についての候補関係を全て既知情報とみなすシステムで，比較手法として設定した．Contains は Fig. 3.6 における step1 のみを行った，すなわち 2 つのフレーズについて片方にもう片方のフレーズが含まれていれば既知情報とみなした場合の結果である．Contains + WMD が提案手法であり，式 (3.10) における単語 i と単語 j の距離 $c(i, j)$ の尺度としてユークリッド距離（euc）とコサイン距離（cos）を用いた場合の結果を示した．Baseline と Contains では 1.4 ポイントの F 値の向上が見られ，WMD による判定を行う提案手法ではさらに 3〜5 ポイントの F 値の向上が見られた．距離尺度はコサイン距離を用いた場合の方が良い結果を示した．しかし，Baseline と Contains + WMD(cos) について t 検定（両側）を行ったところ，有意性は認められなかった（$p = 0.054$）．

4.3.3 考察

Contains + WMD(cos) において誤って判別されてしまった例を以下に示す．これらは全て同値関係にあり，既知情報であると判別されるべきだが，提案手法では新情報であると判別されてしまった．

Table 4.6: Performance of novelty discrimination.

System	Prec.	Rec.	F1
Baseline	0.654	0.722	0.686
Contains	0.668	0.735	0.700
Contains + WMD(euc)	0.700	0.772	0.735
Contains + WMD(cos)	**0.709**	**0.779**	**0.743**

- waterfront walkway linking two Hudson towns
- Hudson River Waterfront Walkway
- new waterfront walkway adjacent to Weehawken Cove
- Hudson River Walkway Pavilion at the Weehawken Cove

これらのフレーズには共通している部分が存在し，特に 'walkway' という単語は全てのフレーズに出現している．フレーズに共通する単語が含まれている場合は同値とみなすことにより，判別性能が向上する可能性が示唆される．そこで，Fig. 3.6 における step1 と step2 の間に，以下の step1.5 の処理を追加することを考える．

step1.5　2 つのフレーズをそれぞれ単語の集合 W_1, W_2 だとみなした時に，共通する単語が N 個以上存在する，すなわち，

$$n(W_1 \cap W_2) \geq N \tag{4.6}$$

を満たす時，類似度 $sim = S_I$ を与え，満たさない場合は step2 に移行する．ただし，$n(A)$ は集合 A の要素の個数を表し，常に $S_C > S_I > \psi$ を満たすものとする．例えば，'Alan Sidney Patrick Rickman' と 'Alan Rickman' は式 (4.6) を満たすので，$sim = S_I$ となる．

step1.5 を追加した手法による実験を行った結果，$Prec. = 0.715$, $Rec. = 0.786$, $F1 = 0.749$ を得た．ただし，$N = 2$ とした．これらは Table. 4.6 における

Contains + WMD(cos) の値を上回っており，Baseline との間には t 検定（両側）において有意性を確認することができた（有意水準 5%）．フレーズに共通する単語が含まれている場合を考慮することは，新情報の判別に有効であったと考えられる．

4.3.4 新情報判別における閾値

提案手法では，2 つのフレーズの抽出元となった文の WMD と，式 (3.12) で表される閾値 ψ との大小により，新情報であるかどうかが決定される．4.2.7 節と同様に，この閾値 ψ を変化させた時の F 値の変化を Fig. 4.5 に示す．閾値の増加に伴って徐々に F 値が上昇していき，$\psi = -0.28$ の時に F 値が最大となった．また，閾値を $\psi = -0.39$ 以下に設定した場合は，Baseline と同等の性能となってしまうことも読み取れる．関係抽出における閾値 ϕ と同様に，新情報判別においても適切な閾値を設定することが提案手法においては重要であることが示唆される．

4.4 定性的な評価

本実験では，提案手法により実際のウェブ文書から関係抽出を行い，直感的に正しい結果を得られているかを定性的に評価することを目的とする．評価に用いるウェブ文書として，英国俳優のアラン・リックマンの死を知らせる英語のニュースステキスト [3] を使用した．このニュースステキストの HTML ソースコードを取得し，HTML タグを除去したものから関係抽出を行った．

実験結果を Table. 4.7 に示す．Titles と AwardsWon については，正しい関係を抽出することができたと言える．しかし，DateOfDeath は誤った日時を抽出してしまった．この原因として，ニュースステキストの中に死没日時が出てこなかったために，確率が低いにもかかわらず（0.271 であった）関係のない日付が死没日時として抽出されてしまった可能性がある．また，ニュースステキストの

[3] http://www.nbcnews.com/pop-culture/celebrity/alan-rickman-british-actor-known-harry-potter-role-has-died-n496346

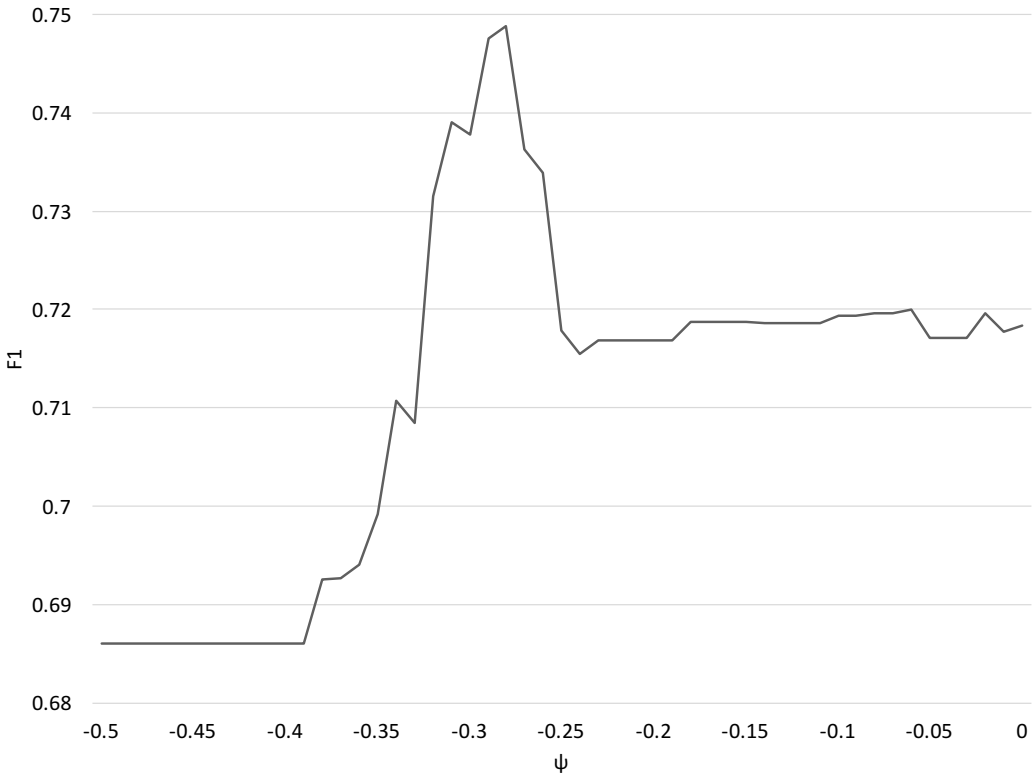


Fig. 4.5: F1 measure vs. threshold ψ for novelty discrimination.

中には‘It said the star had been suffering from cancer.’という一文が存在し，この文から CauseOfDeath として‘cancer’が抽出されるべきであった．抽出されなかった原因として，この文ではアラン・リックマンを表す表記が‘star’であり，これをアラン・リックマンと紐付ける必要がある．そこで，‘star = Alan Rickman’として再度関係抽出を行った結果，正しく抽出されることが確認できた．関係抽出の更なる性能向上のためには，‘star = Alan Rickman’のような紐付けを行うことが重要となる．

さらに，抽出できた関係のうち Titles と AwardsWon に関しては，アラン・リックマンが亡くなった時には既知の情報である．そこで，提案手法により新情報の判別を行った．判別のために利用した抽出元の文を以下に示す．既知の情報としては，アラン・リックマンが亡くなる前日の 1 月 13 日時点での英語版

Table 4.7: Qualitative evaluation of proposed method for relation extraction.

Phrase	Relation
a wonderful actor	Titles
Emmy	AwardsWon
1996	DateOfDeath

Wikipedia のアラン・リックマンの記事に存在した文を使用した．また，判別器の閾値は最適値である $\psi = -0.28$ を用いた．

- Titles
  既知の文: Alan Sidney Patrick Rickman (born 21 February 1946) is an English actor.
  新たな文: Actor David Morrissey also expressed his shock , calling Rickman “ a wonderful actor and lovely man . ”

- AwardsWon
  既知の文: In 1995, he was awarded the Golden Globe, Emmy Award and Screen Actors Guild Award for his portrayal of Rasputin in Rasputin: Dark Servant of Destiny.
  新たな文: His Hollywood star turn came after Rickman played the villain in “ Die Hard , ” and the actor went on to win an Emmy award and Golden Globe in 1996 for his portrayal of Rasputin .

新情報の判別を行った結果，AwardsWon に関しては類似度の値が $-0.228 > \psi$ となり，既知情報であると判別することができた．一方で，Titles については類似度の値が $-0.291 < \psi$ となり，新情報であると判別されてしまった．この原因として，‘a wonderful actor’ の抽出元の文がツイートであることがあげられる．ツイートは 140 文字以内という制限があるため，略語やネットスラングと呼ばれるインターネット上特有の言葉が多用される傾向にある．そのため，Wikipedia の記事データで学習を行った Skip-gram モデルでは，単語の意味を正しく捉えることができなかったことが考えられる．

第5章　結論

本論文では，ウェブ文書からなる時系列テキストストリームに対して関係抽出を行う，ストリーム型の関係抽出手法を提案した．Distant supervision の枠組みを使用することで，ラベル無しコーパスから大量のラベル付きの文を獲得し，それらを学習データとして関係抽出器を学習した．従来の語彙・統語素性だけでなく，Skip-gram モデルによる単語のベクトル表現を意味素性として用いることで，崩れた英文などのノイズの多いウェブ文書の特徴を考慮した．さらに，抽象的関係を導入することにより誤抽出を防ぎ，より頑健な関係分類器が学習できるよう考慮した．また，過去に抽出されたフレーズの意味や抽出元の文脈の情報を用いて，抽出された関係情報が新情報であるかを判別する手法を提案した．

評価実験では，2013 年の TREC KBA SSF タスクの設定に基づいて関係抽出を行い，提案手法が時系列テキストストリームに対する関係抽出タスクにおいて有効であることを示した．意味素性は抽象的関係との組み合わせにより，ウェブ文書からの関係抽出において特に有効であることを確認した．新情報判別においては，提案手法に加えてフレーズの共通部分を考慮することにより，ベースラインとの有意性を示した．しかし,「共通部分がある ⇒ 既知情報である」というように一意に判別を行うため，さらなる性能の向上にはこれを類似度のような指標で表すことが必要である．また，定性的な評価を行い，提案手法により直感的にも正しい結果を得られることを示せた一方で，coreference resolution を導入しなければ正しく関係抽出を行えない事例が存在することが明らかとなった．

今後の課題として，coreference resolution を導入することで，複数の文の情報を統合して関係抽出を行うことを検討している．これにより，より表現力の高い素性を獲得することが期待できる．また，提案手法は閾値の設定が重要であるため，その決定方法を確立することで抽出精度の向上が期待できる．

謝辞

本研究を行う機会を与えて下さり，研究生活全般にわたるご支援を賜りました神戸大学大学院システム情報学研究科の上原邦昭教授に感謝いたします．ご多忙の中，貴重な時間を割いて本論文の審査をお引き受け下さいました同研究科の羅志偉教授，中村匡秀准教授に感謝いたします．日頃より丁寧なご指導を賜り，本論文の執筆にあたり豊富な知識をもって多くのご助言をいただきました甲南大学知能情報学部の関和広准教授に感謝いたします．また，本研究に関する作業補助を頂きました，同学部の４回生鞍野哲也氏，松村侑貴氏に感謝いたします．

参考文献

[1] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction for the web,", In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2670–2676, (2007).

[2] D. Borthakur. "The hadoop distributed file system: Architecture and design,", (2007).

[3] S. Brin, "Extracting patterns and relations from the world wide web,", In *Proceedings of the World Wide Web and Databases International Workshop*, pp. 172–183, (1999).

[4] R. C. Bunescu and R. J. Mooney, "A shortest path dependency kernel for relation extraction,", In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 724–731, (2005).

[5] J. R. Curran, T. Murphy, and B. Scholz, "Minimising semantic drift with mutual exclusion bootstrapping,", In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pp. 172–180, (2007).

[6] H. T. Dang, D. Kelly, and J. J. Lin, "Overview of the TREC 2007 Question Answering Track,", In *Proceedings of the Text REtrieval Conference (TREC)*, (2007).

[7] J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell, "Retrieval and feedback models for blog feed search,", In *Proceedings of the 31st annual*

international ACM SIGIR conference on Research and development in information retrieval, pp. 347–354, (2008).

[8] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification,", *The Journal of Machine Learning Research*, Vol. 9, pp. 1871–1874, (2008).

[9] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta, "Semantically enhanced Information Retrieval: an ontology-based approach,", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 9, No. 4, pp. 434–452, (2011).

[10] J. R. Frank, S. J. Bauer, M. Kleiman-Weiner, D. A. Roberts, N. Tripuraneni, C. Zhang, C. Re, E. Voorhees, and I. Soboroff, "Evaluating Stream Filtering for Entity Profile Updates for TREC 2013 (KBA Track Overview),", In *Proceedings of the Text REtrieval Conference (TREC)*, (2013).

[11] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Ré, and I. Soboroff, "Building an entity-centric stream filtering test collection for TREC 2012,", In *Proceedings of the Text REtrieval Conference (TREC)*, (2012).

[12] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, E. Voorhees, and I. Soboroff, "Evaluating Stream Filtering for Entity Profile Updates in TREC 2012, 2013, and 2014,", In *Proceedings of the TExt Retrieval Conference (TREC)*, (2014).

[13] T. Khot, C. Zhang, S. Natarajan, C. Re, and J. Shavlik, "Bootstrapping Knowledge Base Acceleration,", In *Proceedings of the Text REtrieval Conference (TREC)*, (2013).

[14] M. J. Kusner, E. Y. Sun, E. N. I. Kolkin, and W. EDU, "From Word Embeddings To Document Distances,", In *Proceedings of the International Conference on Machine Learning (ICML)*, (2015).

[15] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit,", In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, (2014).

[16] S. McDonald and M. Ramscar, "Testing the distributional hypothesis: The influence of context on judgements of semantic similarity,", In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pp. 611–616, (2001).

[17] R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge,", In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 233–242, (2007).

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality,", In *Advances in neural information processing systems*, pp. 3111–3119, (2013).

[19] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data,", In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pp. 1003–1011, (2009).

[20] H. Nguyen, Y. Fang, S. Gade, V. Mysore, J. Hu, S. Pandit, A. Srinivasan, and M. Jiang, "A Pattern Matching Approach to Streaming Slot Filling,", In *Proceedings of the Text REtrieval Conference (TREC)*, (2013).

[21] M. S. Nia, C. Grant, Y. Peng, D. Z. Wang, and M. Petrovic, "University of Florida Knowledge Base Acceleration Notebook,", In *Proceedings of the Text REtrieval Conference (TREC)*, (2013).

[22] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora,", In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50, (2010).

[23] E. Riloff, R. Jones, et al., "Learning dictionaries for information extraction by multi-level bootstrapping,", In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, pp. 474–479, (1999).

[24] A. Singhal. "Introducing the Knowledge Graph: things, not strings - Google Official Blog,", `https://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html`.

[25] M. Surdeanu and M. Ciaramita, "Robust information extraction with perceptrons,", In *Proceedings of the NIST Automatic Content Extraction Workshop (ACE)*, (2007).

[26] M. Surdeanu and H. Ji, "Overview of the english slot filling track at the tac2014 knowledge base population evaluation,", In *Proceedings of the Text Analysis Conference (TAC)*, (2014).

[27] Y. Xu, G. J. Jones, and B. Wang, "Query dependent pseudo-relevance feedback based on wikipedia,", In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 59–66, (2009).

[28] C. Zhang, W. Xu, R. Liu, W. Zhang, D. Zhang, J. Ji, and J. Yang, "PRIS at TREC2013 Knowledge Base Acceleration Track,", In *Proceedings of the Text REtrieval Conference (TREC)*, (2013).

[29] M. Zhang, J. Zhang, J. Su, and G. Zhou, "A composite kernel to extract relations between entities with both flat and structured features,", In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 825–832, (2006).

付 録A　ストップワードの一覧

以下に本論文で使用したストップワードの一覧を記しておく．なお，このストップワードはプログラミング言語 Python のモジュール nltk.corpus.stopwords に含まれているものである．

i, me, my, myself, we, our, ours, ourselves, you, your, yours, yourself, yourselves, he, him, his, himself, she, her, hers, herself, it, its, itself, they, them, their, theirs, themselves, what, which, who, whom, this, that, these, those, am, is, are, was, were, be, been, being, have, has, had, having, do, does, did, doing, a, an, the, and, but, if, or, because, as, until, while, of, at, by, for, with, about, against, between, into, through, during, before, after, above, below, to, from, up, down, in, out, on, off, over, under, again, further, then, once, here, there, when, where, why, how, all, any, both, each, few, more, most, other, some, such, no, nor, not, only, own, same, so, than, too, very, s, t, can, will, just, don, should, now

付 録B　Target Relationの定義とWikipedia Infoboxとの対応一覧

本論文の実験の対象となった関係 (Target Relation) の定義を以下に示す．また，Wikipedia の Infobox に存在する関係情報との対応表を Table. B.1 に示す．

Titles　エンティティ（人物）の職業，またはその職業における地位．

AwardsWon　エンティティ（人物）が獲得した賞．

CauseOfDeath　エンティティ（人物）の死亡原因．

DateOfDeath　エンティティ（人物）の死没日時．

EmployeeOf　エンティティ（人物）の勤め先．

TopMembers　エンティティ（組織）の重役．

FoundedBy　エンティティ（組織）の設立者・設立組織．

Affiliate　エンティティと何らかの関係にある別のエンティティ．

AssociateOf　エンティティ（人物）と何らかの関係にある人物．

Table B.1: List of target relations and corresponding Infobox.

Relation	Infobox
Titles	occupation, profession
AwardsWon	awards, prizes
CauseOfDeath	death_cause
DateOfDeath	death_date
RelatedPerson	spouse, children, partner, opponents
RelatedLocation	nationality, birth_place, death_place, residence, alma_mater, resting_place, home_town, body_discovered
RelatedOrganization	employer, organization, boards
RelatedDate	birth_date

付 録C　Target entitiesの一覧

本論文の実験の対象となったエンティティの一覧を以下に示す．エンティティはWikipediaまたはTwitterのURLで与えられる．Table. C.1は4.2節の実験で対象となったエンティティである．また，4.3節の実験ではTable. C.1のエンティティに加えて，Table. C.1のエンティティも対象となっている．表中の「type」はエンティティの種類を表す．種類ごとに対象となる関係が決まっており，以下のように定義されている．

- PER（人物）：Titles, AwardsWon, CauseOfDeath, DateOfDeath, EmployeeOf, Affiliate, AssociateOf
- ORG（組織）：TopMembers, FoundedBy, Affiliate
- FAC（施設）：Affiliate

なお，Wikipediaの URLで定義されているエンティティは表記をリダイレクト情報から獲得し，Twitter場合はその正式名称とした．

Table C.1: List of target entities of the experiment in section 4.2.

No	Entity	type
1	http://en.wikipedia.org/wiki/Barbara_Liskov	PER
2	http://en.wikipedia.org/wiki/Clark_Blaise	PER
3	http://en.wikipedia.org/wiki/Ed_Bok_Lee	PER
4	http://en.wikipedia.org/wiki/Fernando_J._Corbato	PER
5	http://en.wikipedia.org/wiki/Frank_Winters	PER
6	http://en.wikipedia.org/wiki/Geoffrey_E._Hinton	PER
7	http://en.wikipedia.org/wiki/Haven_Denney	PER
8	http://en.wikipedia.org/wiki/Joey_Mantia	PER
9	http://en.wikipedia.org/wiki/Joshua_Boschee	PER
10	http://en.wikipedia.org/wiki/Judd_Davis	PER
11	http://en.wikipedia.org/wiki/Maurice_Fitzgibbons	PER
12	http://en.wikipedia.org/wiki/Paul_Marquart	PER
13	http://en.wikipedia.org/wiki/Richard_Edlund	PER
14	http://en.wikipedia.org/wiki/Ruben_J._Ramos	PER
15	http://en.wikipedia.org/wiki/Sara_Bronfman	PER
16	http://en.wikipedia.org/wiki/Shafi_Goldwasser	PER
17	http://en.wikipedia.org/wiki/Travis_Mays	PER
18	https://twitter.com/AlexJoHamilton	PER
19	https://twitter.com/BobStovall	PER
20	https://twitter.com/KentGuinn4Mayor	PER
21	https://twitter.com/tonyg203	PER

Table C.2: List of added target entities of the experiment in section 4.3.

No	Entity	type
22	http://en.wikipedia.org/wiki/Angelo_Savoldi	PER
23	http://en.wikipedia.org/wiki/Appleton_Museum_of_Art	FAC
24	http://en.wikipedia.org/wiki/Blair_Thoreson	PER
25	http://en.wikipedia.org/wiki/Bob_Bert	PER
26	http://en.wikipedia.org/wiki/Carey_McWilliams_(marksman)	PER
27	http://en.wikipedia.org/wiki/Cementos_Lima	ORG
28	http://en.wikipedia.org/wiki/Charles_Bronfman	PER
29	http://en.wikipedia.org/wiki/Dunkelvolk	ORG
30	http://en.wikipedia.org/wiki/Edgar_Bronfman,_Sr.	PER
31	http://en.wikipedia.org/wiki/Grana_y_Montero	ORG
32	http://en.wikipedia.org/wiki/Great_American_Brass_Band_Festival	ORG
33	http://en.wikipedia.org/wiki/Gretchen_Hoffman	PER
34	http://en.wikipedia.org/wiki/Gwenaelle_Aubry	PER
35	http://en.wikipedia.org/wiki/Intergroup_Financial_Services	ORG
36	http://en.wikipedia.org/wiki/Jamie_Parsley	PER
37	http://en.wikipedia.org/wiki/Jennifer_Baumgardner	PER
38	http://en.wikipedia.org/wiki/Jeremy_McKinnon	PER
39	http://en.wikipedia.org/wiki/Joshua_Zetumer	PER
40	http://en.wikipedia.org/wiki/Ken_Freedman	PER
41	http://en.wikipedia.org/wiki/Luz_del_Sur	ORG
42	http://en.wikipedia.org/wiki/Marion_Technical_Institute	FAC
43	http://en.wikipedia.org/wiki/Mark_SaFranko	PER
44	http://en.wikipedia.org/wiki/Matt_Witten	PER
45	http://en.wikipedia.org/wiki/Reid_Nichols	PER
46	http://en.wikipedia.org/wiki/SIMSA	ORG
47	http://en.wikipedia.org/wiki/Scotiabank_Peru	ORG
48	http://en.wikipedia.org/wiki/Stevens_Cooperative_School	FAC
49	http://en.wikipedia.org/wiki/Susan_Krieg	PER
50	http://en.wikipedia.org/wiki/Theo_Mercier	PER
51	http://en.wikipedia.org/wiki/Weehawken_Cove	FAC
52	http://en.wikipedia.org/wiki/William_H._Miller_(writer)	PER
53	https://twitter.com/BlossomCoffee	ORG
54	https://twitter.com/CorbinSpeedway	FAC
55	https://twitter.com/FrankandOak	ORG
56	https://twitter.com/GandBcoffee	ORG
57	https://twitter.com/MissMarcel	PER
58	https://twitter.com/RobCaud	PER

質疑応答リスト

1. 羅先生: 提案手法により抽出された情報は，新情報判別器にしかフィードバックされていないが，関係抽出器にもした方がよいのでは.
   A. 今回は関係抽出器の学習においては，既存の大規模テキストコーパスを使用することしか考えておらず，実行時のフィードバックは考慮していない．しかし，関係抽出器へのフィードバックにより抽出精度が向上することは期待できるので，今後取り入れていきたい.

2. 羅先生: 意味素性により「2012」が死没日時に分類されているが，実際には違うのでは.
   A. この例では,「2012」を誤って死没日時と判定してしまった．しかし，他の単語については正しく分類されており，それらを統合した結果は正しいので問題ないと考えている．また，分類結果は確率値として得られるが,「2012」が死没日時である確率は大きくなかったので，死没日時であると断定はできないことをシステムが認識していると言える.

3. 羅先生: 網羅性の値はどのように計算するか.
   A. 網羅性には Recall と呼ばれる指標を用いている．Recall は，全ての正解事例のうちシステムがどれだけ抽出することができたかで計算される.

4. 羅先生: 新情報判別に機械学習を使っているが，時系列ストリームの場合は話題が時間とともに変化するので．強化学習のようなアルゴリズムを用いた方がよいのでは.
   A. 今回は逐次的に再学習を行うような仕組みは利用できていないが，時間とともに変化する情報を考慮することは重要であると考えている．今後はそのような仕組みを取り入れていきたい.

5. 中村先生: 既存手法の精度が非常に悪いが，デタラメにやった方が高い値になるのでは.
   A. 今回のタスクは分類ではなく，膨大な文書中から正しいオブジェクトを抽出する必要があるため，デタラメは通用しない．また，従来の関係抽出手法はバッチ処理を行うことである程度の精度を実現しているが，今回は逐次処理であるために精度が低くなっている.

6. 中村先生: テキストストリームからの抽出では，例えば新たに金本が監督になったという情報が抽出できる一方で，時間の経過とともにそれは廃れた情報となり，始めは事実だった情報もそうではなくなる．このような情報を抽出することは可能か.
   A.「金本が監督を辞任した」という情報を抽出し，金本が監督ではなくなったという情報を得ることは可能．しかし，時間経過によって，その情報が事実と異なるかを推定することはできない.

7. 中村先生: 廃れた情報というのは話題にはならないため，テキストから情報を得るのは困難であり，それがこの手法の限界ではないかと思うが，何か考えはあるか.
   A. この点に関しては，現状では人の手による情報の修正が必要だと考えている.