多言語横断 blog 分析エンジンの開発

貞光九月[†]　乗松潤矢[†]　福富崇博[†]

近年、Web 上のデータから有益な情報を自動的に抽出する分析エンジンへの関心が高まっている。対象言語を単言語に限らず、多言語を横断して分析することで、より多くの情報を獲得することも可能であると考えられ、多言語横断型の分析エンジンが求められている。本稿では、最新の研究成果を組み込んだ新しい多言語横断 blog 分析エンジンについて述べる。具体的には、トピックモデルを用いて文書のトピック構造を捉える評価文書分類法とライセンスフリーの対訳コーパスに基づくフレーズ型機械翻訳システムについて実装を行う。

Development of analysis engine for extracting opinions from cross language blog

Kugatsu Sadamitsu,[†] Junya norimatsu[†]
and Takahiro Fukutomi [†]

There has been a recent swell of interest in analysis engines which is the automatic extraction system of beneficial information in web. Furthemore cross language web corpus have more information than single language web corpus, in the such of case we need cross language analysis engine. We implemented new cross language analysis engines employing the newest developments and introduce each development in this report. In first is sentiment analysis based on topic models for extracting topic structures in the document, In second is phrase-based machine translation system using parallel corpus with free license.

1. はじめに

近年 blog をはじめとするツールの充実により、ユーザーが自ら情報を発信することが容易となり、個人の意見を濃く繁栄する CGM(Consumer Generated Media) が増加している。それと同時に、日々増加している膨大な Web 上の情報を分析し、有用な情報を抽出する分析エンジン[1),2)] に対する要求も高まっている。また、日本語で書かれた blog 記事数が世界で最も多くなった (約 40%) という報告が 2007 年になされたが[☆]、他言語で書かれた情報源からの意見抽出についても最近では研究が進められるようになっている。

我々はこのような状況を踏まえ、blog をはじめとする Web 上のテキストデータを対象とした多言語横断分析エンジンの開発した。本システムは昨今の我々の研究成果を分析エンジンに組み込んでいる。具体的には、「トピック情報を用いた評価文書分類」、「ライセンスフリーの対訳コーパスに基づくフレーズベース機械翻訳システム」の 2 つの研究であり、次節からはそれぞれの研究について概説した後、最後にシステムの構成と実際の動作について述べる。

† 筑波大学システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba

☆ Technorati:http://technorati.com/weblog/2007/04/328.html

2. トピック情報を利用した評価文書分類

2.1 評価文書分類と提案手法の概要

blog 等の比較的自由度の高い文書には、その文書が何について述べられているものかを示すタグが付与されていないことが多いが、本節では評価文書分類の精度向上を目的に、そのようなタグの付与されていない評価文書から自動的にトピックを識別し、そのトピックに特化した評価表現のモデルを獲得する手法について述べる。ある対象に対する評価を含む文書 (評価文書) を、肯定極性・否定極性の 2 値ラベルに分類する評価文書分類[3)4)] は、その対象に対する膨大な評価文書を、簡潔かつ定量的な情報として提示できるという点で有益である。評価文書分類では著者の意見を表す評価表現が重要な判断基準となるが、評価表現は対象のジャンルや、著者、文書のスタイル等、様々な要因によって変動する。本稿ではこれらの変動要因をまとめて「トピック」と呼び、トピックを考慮しながら評価文書分類を行うことを考える。

従来の評価文書分類では、単語または数単語を素性とする単語レベルの識別が一般的であった[3)]。一方、Mao ら[5)] や貞光ら[6)] は文を単位とした CRF や HMM を適用し、文が持つ意見の遷移構造を捉える手法を提

案している。また McDonald ら[7] や池田ら[8] は文に対して重みを付与することによって、各文毎の極性を適切に扱う手法を提案している。これらの手法は従来の単語レベルの識別に対し、文レベルの識別と呼べる。本稿で述べる手法は、文書全体を通じて存在する潜在トピックを識別に反映する文書レベルの識別法と言える。

これまでもトピック情報を用いた評価文書分類に関する研究はいくつかなされてきたが、多くはトピックタグを必要とする教師あり学習が前提であった。しかし、評価文書に対し常にトピックタグが付与されているとは限らず、本稿で述べる教師なし学習での評価文書分類法の必要性は高い。具体的には PLSA[9] を用いて自己組織化的にトピック構造をモデル化し、学習によって得られた PLSA に従って評価表現に対するモデルを獲得するという手法をとる。学習には識別学習の一種である MCE(Minimum Classification Error) 学習[10] を用いるが、その際単語毎の極性についての事後確率を仮定することで、極性間でトピック情報を共有する、より頑健なモデルを構築する。また、学習によって得られたモデルから各トピックにおける特徴的な評価表現を抽出し、トピック依存評価表現辞書を構築することも可能である。評価表現辞書はどの単語がレビュアーの意見を強く反映しているかを示すために有益であるが、トピック依存評価表現辞書を用いることでトピックに応じたより細かな情報の提示が可能になる。

2.2 評価文書分類におけるトピックの利用

評価表現の中には、どのトピックにおいてその表現が現れているかによって、極性が大きく異なる場合がある。例えば以下の 2 文において “sad” の極性は明らかに異なっている。

- “The heroine was finally dead with him. It was really sad.”
- “The computer broke just 3 months after I bought it. It was really sad.”

表 1 は Amazon.com☆からの Positive/Negative レビューデータ各 6,800 文書について、4 つの単語 (sad, boring, comfortable, culture) のそれぞれが 3 つのカテゴリ (Electronics, DVD, Magazines) 内において現れた回数を document frequency で数え上げた結果である。“sad” のように極性がトピックによって変わり得るものの他に、“boring” や “comfortable” は特定のトピックにおいて出現しやすい評価表現であることが確認できる。また “culture” のような名詞の場合でも、“Magazines” カテゴリにおいては明らかに Positive 極性に偏って出現している。実際に “*I wanted to learn all I could about the homes and culture of the victorian period*” や “*I'm really into british culture*” のような肯定的表現として現れる典型的な例は見つかるが、否定的表現においてそのような例は見当たらなかった。このように、トピック情報を持った単語の中にも、極性情報を同時に持つ単語が存在すると言える。

☆ http://www.amazon.com

表 1 Amazon の各カテゴリにおいてトピック依存評価表現の現れた数 (df 値)

	Electronics		DVD		Magazines	
	posi	nega	posi	nega	posi	nega
sad	2	4	16	16	1	6
boring	1	0	8	28	0	10
comfortable	11	5	2	2	2	2
culture	0	0	3	6	12	4

これまでに例示したようなトピックによる評価表現の変動を捉えるには、データにトピックタグが付与されている場合、もっとも単純には同じトピックタグの付与された評価文書集合内においてモデルを作成する方法が挙げられる。一方、トピックタグが付与されていないデータについては、評価している対象物と評価表現との直接的な共起関係をモデル化する手法が考えられるが、単純に共起関係を記憶しようとするとパラメータ数が膨大となってしまう上、スパースネスの問題も生じる。そこで本稿では文書全体の背景にあるトピックを捉え、評価表現がそのトピックにおいて Positive/Negative どちらの極性で用いられやすいのかをモデル化する。トピック構造をモデル化する手法として様々なものが考えられるが、本稿ではトピックモデルの一種である PLSA を用いる。

2.3 トピックモデルを用いた評価文書分類

2.3.1 トピックモデルの概要

本稿ではトピックモデルを用いてトピックタグが付与されていない評価文書から隠れたトピックの特徴を自動的に獲得し、トピック毎の評価表現のモデル化を考える。大域的情報を扱うことのできる言語モデルとしては、キャッシュモデルやトリガーモデルが代表的なモデルであったが、トピックモデルはこれらのモデルのように直接単語対単語の関係をモデル化するのではなく、文書に隠れているトピックと単語との関係をモデル化する。トピックモデルにはユニトピックモデルとマルチトピックモデルといった大きく分けて 2 種類のモデルがあるが、1 つの評価文書を 1 つのトピックにハードクラスタリングすることは過適応を招くおそれがあるため、本稿では特に、マルチトピックモデルの一種である PLSA[9] を用いてモデル化を行う。

2.3.2 Probabilistic Latent Semantic Analysis

PLSA[9] は単語レベルでのユニグラム確率の混合を考えることで、1 つの文書が複数のトピックから成ることを表現する。各ユニグラム確率の混合比はそれぞれの文書において個別に計算される。PLSA は次式で

定義される。

$$P(\boldsymbol{w}|d;\boldsymbol{\beta}) = \sum_{\boldsymbol{t}} \prod_{n=1}^{N_d} \beta_{t_n w_n} \theta_{dt_n} \tag{1}$$
$$= \prod_{v=1}^{V} \left\{ \sum_{t=1}^{T} \beta_{tv} \theta_{dt} \right\}^{c_v}$$

w_n は文書 d 中 n 番目の単語、N_d は文書 d に含まれる総単語数、c_v は文書 d 中に含まれる単語 v の数である。$\boldsymbol{t}$ は文書 d における全単語についての隠れたトピック系列を示し、t_n は w_n がトピック t から生成されたことを示す。またトピックの総数を T、θ_{dt} は文書 d におけるトピック t の混合比、β_{tv} はトピック t における単語 v のユニグラム確率を表す。実際の識別を行う際には PLSA を用いて得られる fold-in 確率を用いてベイズ識別を行う。fold-in 確率は、学習で得られたトピック毎のユニグラムパラメータ β については固定したまま、トピックの混合比 θ に関してのみテストセットで適応することで、1 式における確率を求める。

2.3.3 最尤推定と分類誤り率最小化学習における問題点

前節で述べたトピックモデルの推定には一般的に最尤推定が用いられるが、評価文書分類を行う際には問題が生じる。最尤推定は Positive/Negative それぞれのコーパスに対して行うことになるが、この場合の目的関数は対象とするコーパスの尤度であるため、必ずしも分類精度を高める方向に学習が進まない可能性がある。図 2.3.3 は、Amazon.com の Positive/Negative レビューそれぞれについて PLSA を最尤推定し、混合数を変化させていった場合の fold-in パープレキシティと分類精度の関係を示したものである。図中 Correct-PP が正解ラベルについてのパープレキシティ(Positive ラベルを Positive モデルで評価し、Negative ラベルを Negative モデルで評価する)、Wrong-PP が不正解ラベルについてのパープレキシティを示す。ラベル間のパープレキシティに差があればあるほど識別には優位に働きやすいと考えられるが、混合数を増加させるにつれ、両方のラベルにおけるパープレキシティは減少するものの、その間の差はほとんど変わらず、一方で分類精度は悪化してしまっている。

そこで本節では識別学習 (discriminative learning) の一種である MCE 学習[10] を適用し、トピックモデルによる分類精度向上を図る。MCE 学習における目的関数は誤分類尺度に基づく損失関数であり、sigmoid 関数を用いて以下のように定義される。

$$F(d;\boldsymbol{\alpha}) = \frac{1}{1+\exp(-\eta D(d;\boldsymbol{\alpha}))} \tag{2}$$
$$D(d;\boldsymbol{\alpha}) = -\log p(d|\phi;\boldsymbol{\alpha}) + \log p(d|\bar{\phi};\boldsymbol{\alpha})$$

ここで $\phi, \bar{\phi}$ は正解ラベル及び不正解ラベル、$\boldsymbol{\alpha}$ はモデルパラメータ、η は損失関数の傾きを調整するパラメータで、$D(d;\boldsymbol{\alpha})$ は文書 d の誤分類尺度を表す。パラメータの更新には一般最急降下法 (GPD:General Probablistic Descent)[10] を用いて以下のように行う。

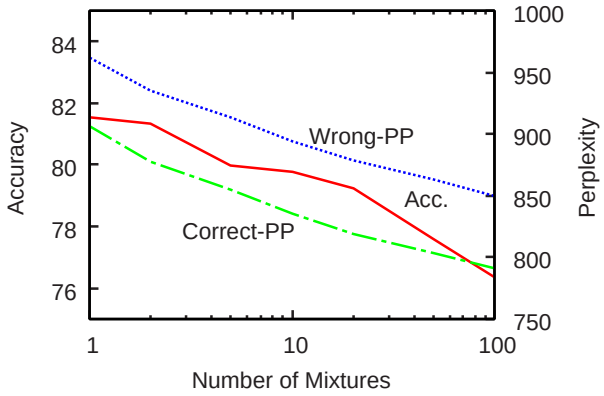

図 1 最尤推定で学習した PLSA による評価文書分類精度とパープレキシティの比較

$$\alpha'^{new} = \alpha'^{old} + \rho \frac{\partial F}{\partial \alpha'^{old}}$$
$$= \alpha'^{old} - \rho\eta F(1-F)\frac{\partial D}{\partial \alpha'^{old}} \tag{3}$$

ここで α' は各パラメータについての対数を表し、和が 1 の条件がある場合については正規化を行う。PLSA においては $\beta_{tv} = \exp(\beta'_{tv}) / \sum_v \exp(\beta'_{tv})$ とする。ρ は GPD において更新の度合いを調整するパラメータで、経験的に定められる。

しかしながら MCE 学習を行う場合、Positive/Negative 間のトピックが必ずしも同じように学習されるとは限らず、かえって過適応を引き起こす可能性がある。そのため Positive/Negative 間で共通のトピックモデルを一度学習しておいてから、それを初期値として学習を開始するといった手順が考えられるが、学習が進んでいくうちに互いのトピック構造が変わる可能性は捨てきれない上、共通の初期値でスタートさせた場合、MCE 学習では初期値の影響が大きいために学習がうまく進まないという問題も生じる。

別な方法として、θ の事前分布、即ち Latent Dirichlet Allocation(LDA)[11] におけるディリクレ事前分布を Positive/Negative 間で共通にするという方法も考えられるが、事前分布だけを共通とするだけで似たようなトピック構造が得られるとは限らない。

2.3.4 極性事後確率を用いた識別学習

MCE 学習は識別学習を行う上で優れた学習法であるが、前節で述べたように、トピックモデルを適用する場合、トピック情報については Positive/Negative 間でなるべく共有しつつ、その上で評価表現の異なりをモデル化したい。ここで、式 2 における誤分類尺度を文書についての単一極性ラベル ϕ から、単語毎についての極性

ラベル系列 $\boldsymbol{\phi}$ として捉えなおすと、それぞれの文書ラベルの元での尤度は $p(d|\phi) = p(d|\boldsymbol{\phi})$,$p(d|\bar{\phi}) = p(d|\bar{\boldsymbol{\phi}})$ と書ける。ここで $\boldsymbol{\phi}, \bar{\boldsymbol{\phi}}$ は全ての極性ラベル系列が ϕ(正解) だった場合と全て $\bar{\phi}$(不正解) だった場合を指す。この単語ラベル系列を用いて誤分類尺度をさらに以下のように変形することができる。

$$
\begin{aligned}
D(d) &= -\log \frac{p(d,\boldsymbol{\phi})}{p(\boldsymbol{\phi})} + \log \frac{p(d,\bar{\boldsymbol{\phi}})}{p(\bar{\boldsymbol{\phi}})} \qquad (4)\\
&= -\log \sum_{\boldsymbol{t}} p(d,\boldsymbol{t}) p(\boldsymbol{\phi}|d,\boldsymbol{t}) + \log \sum_{\boldsymbol{t}} p(d,\boldsymbol{t}) p(\bar{\boldsymbol{\phi}}|d,\boldsymbol{t})\\
&= -\log \sum_{\boldsymbol{t}} p(d,\boldsymbol{t}) \prod_n \mu_{\phi t_n w_n} + \log \sum_{\boldsymbol{t}} p(d,\boldsymbol{t}) \prod_n \mu_{\bar{\phi} t_n w_n}
\end{aligned}
$$

式中 1 行目から 2 行目の変形には、全単語ラベル系列の事前確率を等しいとする仮定 $p(\boldsymbol{\phi}) = p(\bar{\boldsymbol{\phi}})$ を用いている。式中 $\mu_{\phi t_n w_n} = p(\phi|t_n, w_n)$ は n 番目の単語 w_n の潜在トピックが t_n だった場合、その単語の極性ラベルが ϕ である事後確率を示し、以下、極性事後確率と呼ぶこととする。ここで $p(d,\boldsymbol{t})$ が $\boldsymbol{\phi}$ と独立になっていることに注意されたい。$p(d,\boldsymbol{t})$ は単語毎のトピック系列と文書の同時確率を表すが、極性ラベルに対して独立が仮定できるため、ここには Positive/Negative が共通なトピックモデルを持ち込むことができる。以下、極性事後確率を用いた誤分類尺度を PLSA について適用していく。

2.3.5 極性事後確率の PLSA への適用

PLSA の場合は $p(d,\boldsymbol{t})$ を求めることができないため fold-in 確率を用いて極性事後確率を適用する。4 式は PLSA の fold-in 確率を仮定すると以下のように変形できる。

$$
\begin{aligned}
D &= -\log \sum_{\boldsymbol{t}} \{\prod_n \beta_{t_n w_n} \theta_{t_n}\} \{\prod_n \mu_{\phi t_n w_n}\}\\
&\quad + \log \sum_{\boldsymbol{t}} \{\prod_n \beta_{t_n w_n} \theta_{t_n}\} \{\prod_n \mu_{\bar{\phi} t_n w_n}\}\\
&= -\sum_n \log \sum_t \beta_{t w_n} \theta_t \mu_{\phi t w_n}\\
&\quad + \sum_n \log \sum_t \beta_{t w_n} \theta_t \mu_{\bar{\phi} t w_n}
\end{aligned}
$$

θ_t を求める際、$\mu_{\phi tv}$ の更新式に含まれる微分項は最終的に以下のようになる。

$$
\frac{\partial D}{\partial \mu'_{\phi tv}} = c_v \frac{p(v|t)\theta_t \mu_{\phi tv}}{\sum_t p(v|t)\theta_t \mu_{\phi tv}} (1 - \mu_{\phi tv})
$$

ここで $\mu'_{\phi tv} = \mu_{\phi tv} / \sum_\phi \mu_{\phi tv}$ である。本モデルにおいて最終的に必要となるパラメータは、トピック識別用の極性間共通 PLSA 用パラメータと、トピック個数分の極性ユニグラムとなる。

2.4 実験と考察

2.4.1 実 験 条 件

評価実験に際し Amazon.com からレビューデータを取得した。Amazon.com のレビューには評点がレビュアーによって既に付与されており、実験では 17 カテゴリ☆のそれぞれにおいて、各評点 200 レビューずつ、計 13,600 レビューを取得した。トピックタグを全て取り除き、評点 5,4 のレビューを Positive レビュー、評点 1,2 のレビューを Negative レビューとし、10 交差検定を行った。素性には document frequency が 20 回以上の 1gram 単語、計 4,051 単語を presence 素性 (0/1 素性) として用いた。レビューのタイトル、及びレビュアー名はレビューデータに含めていない。MCE 学習を行う際には、過適応を緩和するために全データでの平均化を行っている。学習パラメータについては、それぞれ $\eta = 1, \rho = 0.01$ として固定している。トピック識別用の PLSA の学習の際には、過適応緩和のため tempered EM 法を用いて学習を行った。

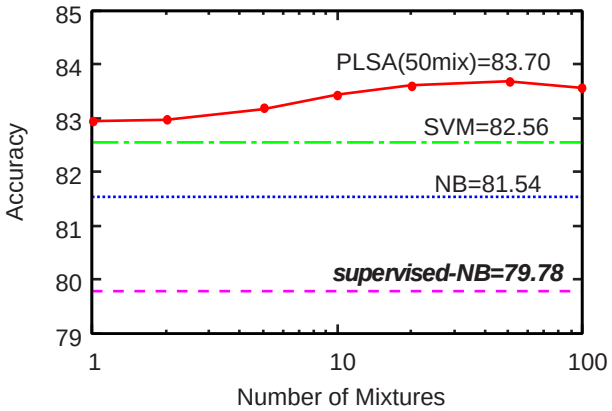

図 2 トピックモデルとベースラインシステムによる評価文書分類 (10 交差検定)

2.4.2 トピックモデルを用いた評価文書分類

トピックモデルを用いた評価文書分類の実験結果を図 2 に示す。ベースラインは Naive Bayes 法 (図中 NB) 及び SVM である。SVM の学習・テストには SVM^{light}☆☆ を用いており、カーネルは線形カーネルである。それに加え、トピックタグ付きデータで学習したトピック別 Naive Bayes モデルに対し、同じくトピックタグ付きのテストセットでテストした場合についても実験を行っている (図中 supervised NB)。過適応を避けるため全トピックを用いた Naive Bayes モデルとの線形補間を行っており、それぞれの Naive Bayes モデルを個別に MCE 学習した後、development データ (各トピック 80 レビューずつ) を用いて線形重みを再度 MCE 学習で求めた。各トピックモデルの μ の初期値にはナイーブベイズ法で得られたユニグラム確率を極性間で正規化して用いている。各モデル名の右の数字は分類精度で、PLSA においては最高精度を示した 50 混合での値を記している。

☆ 現在 Amazon のトップページには 24 個のカテゴリが用意されているが、レビュー数が十分でないことから、うち 7 個のカテゴリについては実験を行っていない

☆☆ http://svmlight.joachims.org/

表 2 提案モデルから得られたトピック依存評価表現の例 (10 混合)

	Tool	Baby	Food	Toy	Movie
Topic	cutting	securely	snack	neighbor	detective
	finger	leaks	texture	lego	violent
	yard	tray	tastes	toys	tony
	neighbor	backpack	protein	boys	director
	pounds	fold	calories	horse	moral
Posi.	needing	arm	awesome	plays	movie
	minutes	dust	amazing	started	season
	decided	complaint	cant	sometimes	us
	available	drawback	color	hours	society
	ago	duty	fall	wait	story
Nega.	short	seat	dont	minutes	moments
	low	rough	sugar	toy	fans
	additionally	pads	save	christmas	knows
	hands	pad	stick	kid	terms
	stop	design	protein	move	enemies
	Book	Computer&Apparel	Software&CD	Service	else
Topic	subscribe	prints	delete	refund	channels
	informative	printers	crashes	emailed	caution
	articles	macbook	restart	responded	displays
	magazines	toe	upgrading	june	locations
	advertisements	epson	server	requested	menus
Posi.	helpful	month	number	service	number
	pages	perfectly	software	replacement	difference
	culture	comfortable	heard	satisfied	mean
	informative	worked	finally	happy	shoot
	journal	problems	song	pleased	material
Nega.	contains	wearing	listen	returned	research
	interesting	replaced	disc	return	etc.
	boring	expected	dvd	broke	problems
	statistics	twice	cd	wash	single
	words	higher	discs	thin	between

PLSA と極性事後確率を用いた提案手法は、1 混合、即ち PLSA を用いていない場合でも、ベースラインである Naive Bayes 法と SVM を上回っている。これは識別学習を行った影響が大きいと考えられる。また PLSA の混合数を増やすことで性能はさらに向上しており、1 混合と 5 混合以上の混合数での交差検定でも 5%の有意水準で精度差が有意であることから、本手法の有効性が示されたと言える。一方で情報量的に有利であるトピックタグ付きの学習・テストデータを用いた supervised NB は、全体の NB モデルとの線形補間を行ったにも関わらず、過適応が生じたために低い精度に留まった。

2.4.3 トピックモデルを用いたドメイン依存辞書

前節で作成したモデルに基づき、ドメイン依存の辞書を作成する。それぞれのモデルにおける極性事後確率 μ から、以下のスコア関数を用いて辞書を作成する。

$$score_{mv} = \frac{\mu_{\phi mv} / \sum_{m'} \mu_{\phi m'v}}{\mu_{\bar{\phi} mv} / \sum_{m'} \mu_{\bar{\phi} m'v}} \quad (5)$$

表 2 は、10 混合 PLSA モデルにおける各コンポーネントにおいて、df が 1000 以下の単語のうち、スコアの高かった順にトップ 5 を示している。“Topic” と記した 2 行目はトピックワードを表し、はじめに学習したトピック識別用 PLSA における各ユニグラム確率 (β) と、単純なユニグラム確率の比における上位 5 単語を示している。また、表中最上段は、トピックワードを見て筆者が名付けたトピック名である。トピックを見ると単純に商品のカテゴリだけでなく、Amazon のサービスについて表しているであろうトピック (表中 “Service”) も現れており、教師なし学習である故の隠れた文書の特徴を捉えることもできている。ただし、“Computer&Apparel” のように複数のトピックが混ざったようなものや、どのトピックにも属さない “else” 等も出現している。これらはトピック数をさらに増やすことで分離が進んでいくが、その一方で似たようなトピックが複数個現れるという傾向も見られた。3 行目、4 行目は肯定・否定評価表現を表している。例えば “Book” トピックにおける “helpful” や “informative” はトピック特有の肯定的表現であり、2.2 節で例に挙げた “culture” も肯定的表現としてモデル化できている。また “Tool” トピックにおける “short” や “stop” は特有の否定的表現であり、どちらの極性においてもトピック特有の評価表現を捉えているといえる。また、“Baby” トピック中の “complaint” は一見すると否定表現のように見えるが、実際のコーパスを見ると “*my only complaint is -*” 等の表現が多く見られ、肯定的文書の中で限定的な批判を行う場合に現れやすい表現となっていた。“Computer&Apparel” の “expected” も同様に、“*I expected much more of the coat than I got*” のような文脈で用いられることが多かった。一方、評価表現として理由付けのできない単語が辞書中に現れていることから辞書構築について改善の余地は残されていると言えるが、トピック情報が付与されていないコーパスからこのようなトピック依存評価表現辞書を自動的に構築する可能性を示せ

たといえる。

3. フレーズを用いた統計的機械翻訳システム

3.1 統計的機械翻訳の概要

ある自然言語を別の言語に自動的に変換する技術は機械翻訳とよばれ、これまで盛んに研究されてきている分野のひとつである。現行の機械翻訳システムは主に人手で翻訳規則を記述する、ルールベースとよばれる手法で開発されている。しかしながら自然言語の規則には例外や曖昧性があるため、ルールベースではこれら全てを網羅するのが困難であり、性能向上は難しい。一方、計算機の発達によって近年研究されてきているのが、統計的な手法を用いた機械翻訳 (以下、統計的機械翻訳) である[12)]。統計的機械翻訳は膨大な対訳コーパス☆から機械学習によって翻訳規則を自動抽出する手法であり、ルールベースに比べて一貫性や網羅性の面で優位であるといえる。統計的機械翻訳では、ベイズの定理に基づいた以下の式によって翻訳がなされる。

$$\begin{aligned}\hat{E} &= \arg\max_{E} P(E|F) \\ &= \arg\max_{E} P(E)P(F|E)\end{aligned} \tag{6}$$

ここで F は翻訳される元の言語 (原言語)、E は翻訳される先の言語 (目的言語)、$\hat{E}$ は、システムが出力する翻訳結果である。この式より、統計的機械翻訳システムには 3 種類の研究分野があると言える。

- 言語モデル $P(E)$:ある文 E がその属する言語の文として生起する確率を与える。
- 翻訳モデル $P(F|E)$:ある文 E が原言語の文 F に翻訳される確率を与える。
- デコーダ $\arg\max_E$:最も翻訳文としての確率が高くなる目的言語の文 E を返す。

なかでも翻訳モデルは翻訳の確からしさを与えるモデルであり、システム全体の翻訳精度に大きな影響をあたえる。

本システムでは、フレーズモデルによる統計的機械翻訳を用いた。ここで、フレーズモデルは数単語を単位として、翻訳の質を評価するモデルのことである。以下ではこのフレーズを用いた統計的機械翻訳について概説した後統計的機械翻訳に必要不可欠なコーパスについて一般公開に耐えうるフリーなコーパスについて検討を行う。

3.2 フレーズを用いた統計的機械翻訳

フレーズモデルを用いた翻訳は以下のように定式化される[13)14)]。原言語文 F と目的言語文 E が I 個のフレーズ対 $\hat{f}_1^I$、$\hat{e}_1^I$ と表せるとすると、翻訳文 $\hat{E}$ は、

$$\hat{E} = \arg\max_{E} P\left(\hat{f}_1^I|\hat{e}_1^I\right). \tag{7}$$

と書ける。ここで、$\hat{f}_i$、$\hat{e}_i$ をそれぞれ原言語、目的言語の i 番目のフレーズであるとして、

$$P\left(\hat{f}_1^I|\hat{e}_1^I\right) = \prod_{i=1}^{I} \phi\left(\hat{f}_i|\hat{e}_i\right) d\left(a_i - b_{i-1}\right) P_w\left(\hat{f}_i|\hat{e}_i\right)^{\lambda} \tag{8}$$

と仮定する。ここで、ϕ はフレーズ翻訳確率、d は歪み確率、a_i は、目的言語のフレーズ $\hat{e}_i$ に対応する原言語のフレーズの最初の単語位置、b_{i-1} は、目的言語のフレーズ $\hat{e}_{i-1}$ に対応する原言語のフレーズの最後の単語位置であるとする (図 3)。λ は適当な重みである。 このとき、フレーズ翻訳確率 ϕ、歪み確率 d、語彙重み P_w を式 9 から式 13 によりそれぞれ定義する。式中 $\mathrm{count}(x, y)$ は任意のフレーズペア x, y がコーパスから抽出された回数である。

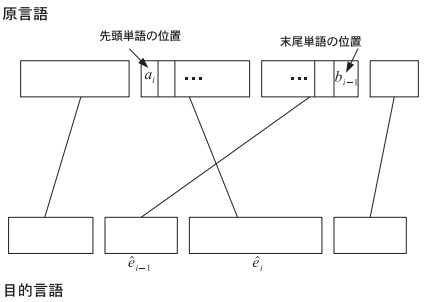

図 3 フレーズ歪みのパラメータ a_i,b_{i-1} の例

$$\phi\left(\hat{f}_i|\hat{e}_i\right) = \frac{\mathrm{count}\left(\hat{f}_i, \hat{e}_i\right)}{\sum_{\hat{f}} \mathrm{count}\left(\hat{f}_i, \hat{e}_i\right)} \tag{9}$$

α を適当な数として、

$$d\left(a_i - b_{i-1}\right) = \alpha^{\left|a_i - b_{i-1} - 1\right|}. \tag{10}$$

ここで、$\boldsymbol{a}$ は単語対応として、語彙重み $P_w(\hat{f}|\hat{e})$ は、以下の語彙翻訳確率分布 $w(f|e)$、単語アラインメント付き語彙重み $P_w(\hat{f}|\hat{e}, \boldsymbol{a})$ を用いて計算する。すなわち、

$$w\left(f|e\right) = \frac{\mathrm{count}\left(f, e\right)}{\sum_{f'} \mathrm{count}\left(f', e\right)} \tag{11}$$

$$P_w\left(\hat{f}|\hat{e}, a\right) = \prod_{i=1}^{n} \frac{1}{|\{j|(i,j) \in \boldsymbol{a}\}|} \sum_{\forall (i,j) \in \boldsymbol{a}} w\left(f_i, e_j\right) \tag{12}$$

とおいて、

$$P_w\left(\hat{f}|\hat{e}\right) = \max_{\boldsymbol{a}} P_w\left(\hat{f}|\hat{e}_i, \boldsymbol{a}\right)^{\lambda} \tag{13}$$

と表現する。

以上がフレーズモデルの詳細である。本プロジェクトではこのモデルを採用した。

3.3 フリーデータによる学習コーパス構築

3.3.1 フリーデータの概説

統計的機械翻訳は人手で翻訳された翻訳例を元にモデルの学習を行うため、コーパスの性質によって、対数の法則により、学習に用いる翻訳例が実システムの目的とする言語の母集団の性質を規定すると考えられ、その翻訳精度に大きな影響が及ぶ。

☆ 翻訳対象となる 2 つの言語それぞれの文と文が、対訳ペアとして一対一に対応づけられたデータ

今回のプロジェクトでは、システムの公開を目指して自由に利用できるデータ☆1(以下、フリーデータ) のんみを用いてシステムを構築し、検証を行った。

本システムでは日英、英独の翻訳システム作成をした。日独間の翻訳については、一度英語に翻訳した後英語をドイツ語に翻訳することによって実現する。現在の統計的機械翻訳においては、対訳コーパスとして、2 言語の文と文が一対一に対応した対訳コーパスを用いている。今回のシステム構築のために用いたデータを表 3 に示す。

内山による対訳コーパス☆2 とは、日本語と英語の両言語で読むことができるフリーな文書などに対して英日間の文対応をとったデータである。英語データは主に、Project Gutenberg☆3、日本語データは主に、プロジェクト杉田玄白☆4から取得されている。ただし、一部に二次配布が禁止されている作品も含まれている。今回はシステムで使用し、可能な限り公開することも考慮しているため、2 次配布が禁止されている、「トラベル英会話」「ことわざ」の 2 作品については除外した。

次に聖書は、日本語訳の著作権が失効している日本聖書協会発行の口語訳聖書第 1 版と、Public Domain の下で翻訳が行われている、World English Bible☆5を用いた。聖書は若干の揺らぎはあるものの、世界でほぼ共通に章と節がふられており、節のレベルを文とみなして対応付けることで、精度の高い対訳を得ることができる。明らかに対応がずれていることが判明した数ヶ所については、手動でこの対応を修正した。また、本来節がわかれるべきところが翻訳の都合上一つになってしまったところについては、対応する言語で節がわかれていても一つにまとめ、学習を行った。

最後に、Wikipedia☆6は多言語の百科事典プロジェクトであり、一部英語から日本語へ翻訳されている記事が存在しすることから、翻訳されている文を抽出し、対応付けることを試みた。Wikipedia からの対訳文抽出法としては、まず、Wikipedia 全記事に対する本文データのダンプを取得し☆7、各項目の対応を取得する。今回は 128011 項目の対応を得た。次に、ノイズとなりうる箇条書きなどを削除したのち、文対応取得には内山による文対応手法[15] を用いた。この手法では、単語の対訳辞書が必要であるため、EDICT☆8を用いた。

☆1 著作権が失効している、または、著作権者が自由な利用を認めているデータのこと

☆2 http://www2.nict.go.jp/x/x161/members/mutiyama/align/index.html

☆3 http://www.gutenberg.org/wiki/Main_Page

☆4 http://www.genpaku.org

☆5 http://ebible.org/bible/web/

☆6 http://www.wikipedia.org/

☆7 ダンプ日は、英語が 2007 年 10 月 18 日、日本語が 2007 年 9 月 23 日のものを用いた

☆8 http://www.csse.monash.edu.au/~jwb/j_edict.html

表 3 日英対訳コーパス情報

	内山による対訳コーパス	聖書	Wikipedia	混合コーパス
日本語語彙数	47,031	15,192	51,198	83,441
英語語彙数	40,629	12,689	43,970	71,615
対訳文数	107,723	29,101	38,126	174,950

表 4 英独コーパス情報

英語翻訳モデル用データ語彙数	65,888
英語言語モデル用データ語彙数	86,700
ドイツ語翻訳モデル用データ語彙数	205,378
ドイツ語原語モデル用データ語彙数	277,199
対応文書数	751,088

この手法では、文対応が評価されたスコアが得られ、その値が閾値を超えている文の対応を対訳として採用した。スコアは二種類存在する。文の対応のみから得たスコアと、文書の対応まで考慮したスコアである。今回は前者を 0.25, 後者を 0.03 に閾値をおいた。この手法により、以下のような対訳が得られた。

- これは彼が最後に発見した小惑星である。
- it was his last asteroid discovery.

また、日本語に関しては、学習の途中で得られたコーパスから文切り用の言語モデルを得た。この本システムではこの言語モデルを用いて翻訳するデータの文切りを行う。

最後に、これらのデータを全て混ぜた学習も行った。表 3 における混合コーパスの欄に詳細を示す。

英独間のフリーの対訳コーパスには Europarl コーパス☆9、を用いた。特に、今回は NAACL 2006 Workshop で用いられたコーパス☆10を使用し、同時に配布されていた単語アラインメントと言語モデルを利用した。コーパスの情報を表 4 に示す。

3.3.2 フリーデータによるフレーズ学習結果

まず、英日の各コーパスについて述べる。それぞれの学習に共通の条件を表 5 に示す。翻訳モデルの学習には moses☆11付属の学習プログラムを用いた。言語モデルの学習には SRI Language Modeling Toolkit☆12を用いた。

学習結果を 6 に示す。英日対訳文コーパスについて、100 万以上のフレーズペアが得られたが、これらのほとんどは “a” が “あっと い うま の” と対応しているといった、誤った対応であった。このコーパスは小説などからデータが取られていることが多小説などは意

表 5 学習条件

言語モデル	5-gram, Modified Kneser-Ney Discounting, 線形補間法を使用
単語アラインメント法	grow-diag-final
歪モデル学習法	msd-bidirectional-fe

☆9 http://www.statmt.org/europarl/

☆10 http://www.statmt.org/wmt06/shared-task/

☆11 http://www.statmt.org/moses/

☆12 http://www.speech.sri.com/projects/srilm/

表 6 フレーズ学習結果

	内山による対訳コーパス	聖書	Wikipedia	混合コーパス
英日方向取得フレーズペア数	1,351,747	1,855,515	613,536	2,364,731
日英方向取得フレーズペア数	1,349,092	1,884,492	603,842	2,358,292

訳されている事が多く、このようなデータから正しく学習するにはより大量のデータが必要になると考えられる。

次に聖書による対訳での学習結果を述べる。聖書は対訳数が少ないにもかかわらず、200 万弱のフレーズペアが得られた。これは、フレーズ対応の質は比較的良いが、語彙が特殊であり、使える場面が限られてくると思われる。また、聖書のデータにはタグや訳注が残っており、ノイズとして対訳の質が悪くなっている可能性も考えられるため、今後改善していきたい。

次に、Wikipedia については、Wikipedia による学習は 60 万程度のフレーズ数となった。また、得られたフレーズの対応も悪いという結果であった。これは、文対応の精度が悪いデータが多く混入してしまい、精度を下げてしまったのではないかと考えられる。Wikipedia のデータについては、スコアが良い今回は約 38,000 対を採用したが、現状のスコアでは低い値が当てられていても、実際はよく対応しているものや、高い値でも、間違った対応のものが見られた。新語などの辞書にない単語により、スコアが不正確なものになっている可能性がある。したがって、新たな辞書を用いて文対応のスコアをつけ直すことにより、対訳抽出をより正確に行うことも今後の課題である。

次に、混合コーパスでの学習結果を示す。学習データが多いことから、フレーズペアの数は最多となったが、依然としてフレーズ対応の質が悪かった。

全体として、現状では満足できるコーパスが得られなかった。これらは主に対訳の精度が悪いことと学習データ量が少ない事が原因として挙げられる。

4. 多言語横断 blog 分析システム「百葉箱」の実装

4.1 システムの構成

図 4 は本システムの処理の流れを図示したものである。図中ユーザーが日本語を使うことを仮定しているが、ユーザーの使う言語が英語・ドイツ語であっても、基本的に処理は同じである。

まず日本語 (ユーザーネイティブ言語) については、ユーザーが入力したクエリによってコーパスに対し検索が行われる。検索にヒットしたそれぞれの文書に対し 2 節で述べた、トピックモデルを用いた評価表現モデルを用いて評価文書分類を行う。文書数の統計を円グラフに表すと同時に、検索された blog 集合の中で頻繁に用いられている評価表現 (図中「軽い」「汚れ」

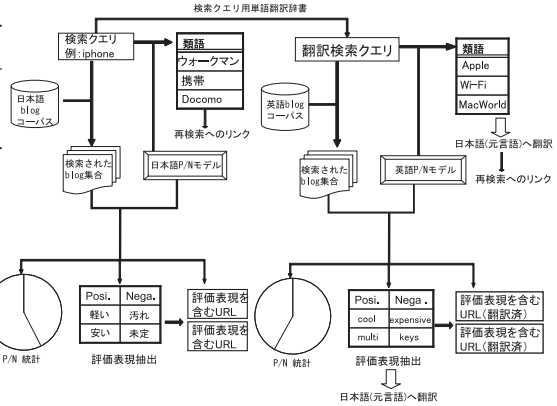

図 4 多言語 blog 分析エンジンのシステムフローチャート (ユーザーの言語に日本語、他言語に英語を仮定)

等) を抽出・提示する。具体的には式 5 を拡張したスコア関数を用いる。

$$score'_v = I(v,q) \sum_m \frac{\beta_{mv}}{p_{uni}(v)} \frac{\mu_{\phi mv} / \sum_{m'} \mu_{\phi m'v}}{\mu_{\bar{\phi} mv} / \sum_{m'} \mu_{\bar{\phi} m'v}} \quad (14)$$

ここで、$I(v,q)$ は単語 v と検索クエリ q との相互情報量を表し、$I(v,q) = \log df(v,q)/(df(v) \cdot df(q))$ で定義する。提示の際、評価表現に対するスコア値の値をフォントの大きさとして反映させている。最後に評価表現を実際に用いている blog へのリンクを生成することで、評価表現側の処理を終える。一方で類語の提示には、クエリとの相互情報量の高い単語、およびトピック識別用 PLSA においてクエリのユニグラム確率の高いとトピックと同じトピックにおいて、同様にユニグラム確率の高くなる単語の中からランダムに選択した単語を、狭義および広義の類似語として提示する (図中「ウォークマン」「携帯」等)。出力された類似語に対しては、再度システムに対してその類似語を検索クエリとして投げるようにリンクを生成する。

次に英語コーパス（ユーザー非ネイティブ言語）に対しては、ユーザーが入力したクエリを、Wikipedia 項目間辞書☆と一般の言語間辞書を用いて翻訳し、英語コーパスに対し検索を行う。英語コーパスで学習されたトピック依存評価表現モデルを用いて、評価文書分類と評価表現抽出を行う処理までは日本語の場合と同様に行い、得られた評価表現を日本語 (ユーザーネイティブ言語) に翻訳する (図中 “cool”,“expensive” ⇒「格好いい」「高い」等)。評価表現からは日本語コーパス同様、blog へのリンクを生成するが、そのリンク先は単なる blog ではなく、システムによって blog の内容が日本語に翻訳された結果を示す。類語についても評価表現の場合と同様に、日本語に翻訳した結果を返している (図中 “Appple”,“Wi-Fi” 等)。

☆ 3.3.1 節で述べた Wikipedia の項目対応を用いて、その項目名同士を対訳辞書として用いた。

表 7　システムに用いる Amazon レビューデータ数統計

star num.	Japanese			English			German		
	All	train	test	All	train	test	All	train	test
star 1	5281	2500	2641	19769	6000	9885	10590	3500	5295
star 2	5626	2500	2813	12931	6000	6466	7596	3500	3798
star 3	13284	-	6642	20968	-	10484	11583	-	5792
star 4	31731	2500	15866	51590	6000	25795	23463	3500	11732
star 5	67693	2500	33847	149728	6000	74864	71473	3500	35737

4.2　システムの実験条件と動作

本稿では blog を分析の対象としてシステムを構築してきたが、システムの公開に向けて著作権等の解決が難しかったため、テストデータには擬似的な blog データとして Amazon のレビューデータを用いることとする。レビューデータに商品名およびタイトルを付与したものを blog の 1 記事とみなしている。Amazon レビューデータを、トピック依存評価表現モデルの学習用と、分析対象となる擬似 blog オープンテスト用の 2 種類に分けて使用した。用いた Amazon コーパスの統計量を表 7 に示す。学習コーパスについては、本稿では事前分布を仮定しないため各評点において同一数と仮定しており、元コーパスを学習用・テスト用に二分した後にランダムサンプリングしているため、テストデータとの間においてレビューデータの重複はない。 また、機械翻訳システムの学習条件は 3 節に述べたものに従うものとする。

図 5 は、本プロジェクトで作成した多言語横断 blog 分析エンジン「百葉箱」の動作のスナップショットである。ただし、本稿執筆時点ではモデルの学習が完了していないものがあるため、図中のテキスト情報は手動で設定している。検索クエリ「iPhone」に対し、トピックとしての関連語、評価の統計情報を示す円グラフ、その根拠となった評価キーワードが全て日本語で示されている。円グラフはポジティブ・ネガティブだけでなく、中立も表示しているが、これは評価表現モデルの出力するポジティブ・ネガティブの確率比に適切な閾値を設定することで実現している。最後に図 6 は英語の評価キーワード「きれい」のリンクをクリックした場合のレビューに対する翻訳イメージを示す。

図 5　多言語 blog 分析エンジン「百葉箱」のスクリーンショット (画像は開発中のもの)

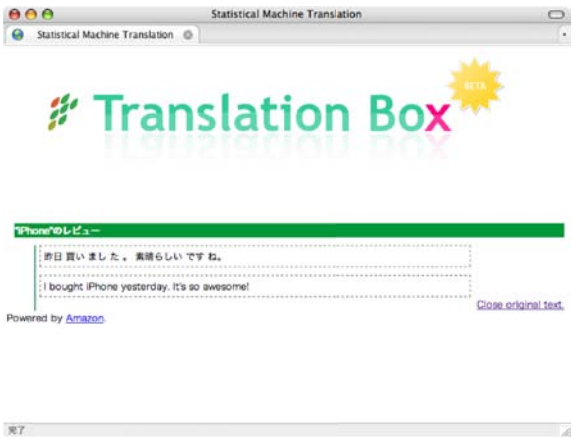

図 6　多言語 blog 分析エンジン「百葉箱」の翻訳リンクにおけるスクリーンショット (画像は開発中のもの)

5. おわりに

本稿では多言語 (日・英・独) を横断して blog からの意見抽出を行うシステムと、そのシステムの実現に必要とされる研究の成果について述べてきた。ほとんど前例のない多言語分析 blog 分析エンジンを一般に公開できるような形にまで開発した成果は大きいと言えるだろう。

今後の課題は、実験の節で述べたように実際の blog に対して本システムを運用することであるが、法的な問題が関係するため、状況の変化を待たざるを得ない部分がある。また、blog データにシステム適用する際に、新たに前処理を加える必要があると考えられる。例えば blog は 1 つの商品についてレビューをしていることはあまりなく、話題は頻繁に転換する。そのため、blog 文書中のどの部分がどの商品について意見を述べているかを定めるという処理を行う必要があるだろう。さらに、レビューと blog の文体は同じではないと考えられるため、評価表現モデルにおける出力結果の精度が悪化することが予想されるため、より汎用性の高いモデルの設計が必要となる。機械翻訳に関しては、Wikipedia を用いて対訳コーパスを増強しているとはいえ、まだ十分とは言いがたい量であり、さらなるコーパスの増強が求められる。

謝　　辞

本研究の一部は，魅力ある大学院教育イニシアティブ「実践 IT 力を備えた高度情報学人材育成プログラム」による．

参 考 文 献

1) Cass, S.: Fountain of knowledge, *IEEE Spectrum*, Vol. 41, No. 1, pp. 68–75 (2004).
2) Nanno, T., Suzuki, Y., Fujiki, T. and Okumura, M.: Automatically Collecting, Monitoring, and Mining Japanese Weblogs, *Proceeding of WWW2004: the 13th international World Wide Web conference* (2004).
3) Pang, B. and Lee, L.: Thumbs up? Sentiment Classification using Machine Learning Techniques, *Proceedings of the Conference on Empirical Methods in Natural Language processing(EMNLP)*, pp. 76–86 (2002).
4) 乾孝司, 奥村学: テキストを対象とした評価情報の分析に関する研究動向, 自然言語処理学会論文誌, Vol. 13, No. 3, pp. 201–241 (2006).
5) Mao, Y. and Guy, L.: Isotonic Conditional Random Fields and Local Sentiment Flow, *Neural Information Processing Systems*, Vol.18 (2007).
6) 貞光九月, 山本幹雄: 文を単位とする文書構造を用いた評価文書分類, 言語処理学会第 13 回年次大会, pp. 230–233 (2007).
7) McDonald, R., Hannan, K., Neylon, T., Wells, M. and Reynar, J.: Structured Models for Fine-to-Coarse Sentiment Analysis, *the 45th Annual Meeting of the Association for Computational Linguistics(ACL-2007)*, pp. 432–439 (2007).
8) 池田大介, 高村大也, Ratinov, L.-A., 奥村学: 単語極性反転モデルによる評価文分類, 情報処理学会研究報告,NL-180, pp. 43–48 (2007).
9) Hofmann, T.: Probabilistic Latent Semantic Indexing, *Proc. of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, Berkeley, California, pp. 50–57 (1999).
10) Juang, B.-H. and Katagiri, S.: Discriminative learning for minimum error classification, *IEEE Trans. Signal Processing*, Vol.40, pp.3043–3054 (1992).
11) Blei, D., Ng, A. and Jordan, M.: Latent Dirichlet allocation, *Journal of machine Learning Research 3* (2003).
12) Brown, M., Hughey, R., Krogh, A., Mian, I., Sjolander, K. and Haussler, D.: Using Dirichlet Mixture Priors to Derive Hidden Markov Models for Protein Families, *Intelligent Systems for Morecular Biology* (1993).
13) Koehn, P., Och, F. J. and Marcu, D.: Statistical phrase-based translation, *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Morristown, NJ, USA, Association for Computational Linguistics, pp. 48–54 (2003).
14) Koehn, P.: *Pharaoh a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models User Manual and Description for Version 1.2* (2004).
15) Utiyama, M. and Isahara, H.: Reliable Measures for Aligning Japanese-English News Articles and Sentences, *ACL-2003*, pp. 72–79 (2003).