2004－SLP－50　(11)
2004／2／7

自然発話文における統計的な意図理解手法の検討

白木 将幸[†]　伊藤 敏彦[‡]　甲斐 充彦[§]　中谷 広正[‡]

要 旨

近年、音声認識技術の進展に伴い、音声対話システムの研究が盛んに行われている。しかしながら、音声認識の誤認識や話し言葉特有の表現などが原因で、自然発話においてユーザの意図を正確に抽出することは難しく、十分な結果が得られていない。そこで我々は、自然発話の意味理解において、統計的な意図推定を用いた方法を提案する。これは、発話の大まかな意図というものを捉えて、それをもとに部分的な解析を行うことで発話の意味を理解する手法である。意図の推定規則はコーパスから学習して獲得するが、実際の音声認識結果や自然発話を含むコーパスを用いることで、頑健な意図理解が可能となる。本論文では、意図理解に、N-gram モデル、ベクトル空間モデル、Support Vector Machine(SVM) を利用し、意図推定精度における実験と評価を行った。

Investigation of statistical techniques for intention understanding in spoken dialog

Masayuki Shiraki[†]　Toshihiko Itoh[‡]　Atsuhiko Kai[§]　Hiromasa Nakatani[‡]

Abstract

Recently, research on spoken dialog systems has been active with progress of the speech recognition technology. However, it is difficult to extract user intention correctly from natural utterance. Most of these difficulties are due to the errors of speech recognition results, and a variety of linguistic phenomena included in natural utterance. We propose statistical methods to extract user intention from natural utterance. By learning examples, a set of rules which are robust to various linguistic phenomena can be automatically acquired. In this paper, N-gram model, vector space model, and Support Vector Machine (SVM) are used for understanding user intention. We perform the experiments of intention understanding and evaluate the performances of those methods.

1　はじめに

近年、音声認識技術の進歩に伴い、音声対話システムの研究も盛んになり、実用的なシステムもいくつか開発されるようになった [1][2]。

音声対話システムの構築には、非音声の対話システムの応用が考えられるが、初期の対話システムでは、文法に基づいた解析型の手法を用いていたため、非文法的発話の意味を理解することは難しい問題があった。最近では統計的手法や機械学習を用いた意味理解手法が提案されるようになり、頑健な解析手法が利用されるようになってきた。しかし、それらの手法を用いた対話システムでも、自然発話による対話で満足のいく性能を得ることはできていない。これは、音声認識結果に誤りが含まれることや、自然発話の多様な表現 (間投詞、言い誤りなど) が原因として考えられる。そのため、音声対話システムの構築にはこれらの問題にうまく対応した手法の提案が求められている。

そこで本研究では、発話の意味理解において、統計的な意図推定を用いた方法を提案する。意図とは、発話の大まかな意図のことであり、それを推定してから部分的解析を行うことで発話の意味を理解

[†]静岡大学大学院 情報学研究科 情報学専攻
[†]Graduate School of Information, Shizuoka University
[‡]静岡大学 情報学部
[‡]Faculty of Information, Shizuoka University
[§]静岡大学 工学部
[§]Faculty of Engineering, Shizuoka University

する。音声認識誤りや話し言葉特有の表現をコーパスから学習することで、頑健な解析が期待できる。

また、この手法を用いても自然発話の意味を100%に近い精度で理解することは困難であるが、目指す対話システムの目的は対話によるタスクの達成である。意図を推定できれば意味理解に多少の誤りを含んでも応答や対話制御によってそれらを修正することは可能だと考えられる。そのため我々は、自然発話の意味理解精度を追求するのではなく、円滑な対話が行えるシステム構築を目指す。

本論文では、この手法において発話の意図推定がどの程度行えるかを実験・評価した。学習方法として、N-gram モデル [3]、ベクトル空間モデル [4]、Support Vector Machine(SVM)[5] を利用し、各手法において精度の評価実験を行った。

2 発話の意味理解

本研究における意味理解手順と「発話意図」に付いて以下に述べる。

2.1 意図理解の手順

本研究における意味理解の手順を図 1 に示す。各処理の概要は以下に示す通りである。また、この処理によって得られる意味理解結果の例 (ホテル検索・予約タスクによる発話) を図 2 に示す。

1. **音声認識**: 発話の音声データをテキストデータに変換する。変換されたテキストデータが次の処理に使われるが、音声認識結果には誤りを含む場合がある。その場合は誤りを含んだテキストデータが渡される。
2. **形態素解析**: 音声認識結果により得られたテキストデータを形態素に分割する。
3. **発話の意図単位分割**: 得られた発話 (形態素列) が複数の意図を含むと考えられる場合、次の意図理解処理がうまく行えない。そのため、前処理として発話の分割を行う。発話の分割処理もコーパスから学習した分割規則を利用することで自動的に行う。
4. **意図理解**: 分割された全発話から「発話意図」(2.2 節) を推定する。これは統計的手法により得られた意図推定規則を用いて行う。得られた発話意図は、各々の発話データに付与され、次の処理に用いる。
5. **部分的解析**: 意図が付与された発話データからその意図に基づいて必要な情報を抽出する。解析には部分的な構文解析やキーワード解析などを用いる。

2.2 発話意図

人工知能学会の談話タグワーキンググループによる談話行為タグ [6] に基づき、タスク達成に必要と思われるものをタスク依存意図として拡張した。

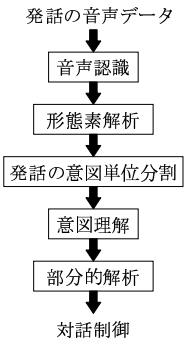

図 1: 発話の意味理解手順

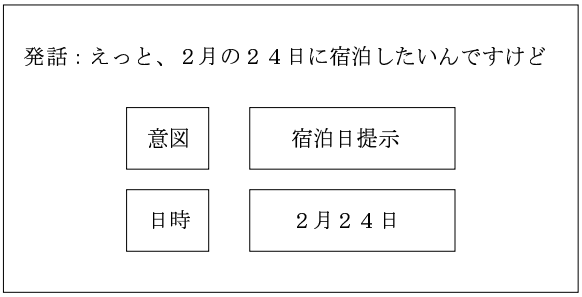

図 2: 発話の意味理解結果の例

発話意図は次のような特徴を持つ。

- 発話の発話内行為を明示的に示す。
- 談話行為タグに比べ、タスクに依存した下位レベルの意図を表現する。

発話意図は全部で 79 種類である。その一部を表 1 に示す。

表 1: 発話意図

談話行為タグ	タスクに依存した意図
依頼	ホテル検索依頼
希望	宿泊日・期間提示
	人数提示
	料金提示
情報伝達	ホテル情報伝達
未知情報応答	場所応答
	人数応答
	ユーザ名応答

3 統計的な学習

意図推定方法として、N-gram モデル、ベクトル空間モデル、SVM を用いた方法を以下に示す。

3.1 N-gram モデル

N-gram モデルにおいて、単語列 W が意図 I である確率は、

$$P(I|W) = \frac{P(I)(W|I)}{P(W)} \quad (1)$$

より、推定意図 $\hat{I}$は式 (2) のように等価に表される

$$\begin{aligned} \hat{I} &= \underset{I}{argmax}\, P(I)P(W|I) \\ &= \underset{I}{argmax}\, P(I)\prod^{n} P(w_n|I, w_{n-1}, \ldots, w_1) \end{aligned} \quad (2)$$

ここで、w_n とは単語列 W の n 番目の単語を表す。N-gram モデルは、ある時点での単語の生起は直前の (N-1) 単語にのみ依存すると考えている。そのため N=1(uni-gram) の場合は 式 (2) の $P(w_n|I, w_{n-1}, w_{n-2}, \ldots, w_1)$ の部分を $P(w_n|I)$ と置き換えることができる。同様に N=2(bi-gram) の場合は $P(w_n|I, w_{n-1})$、N=3(tri-gram) の場合は $P(w_n|I, w_{n-1}, w_{n-2})$ と置き換えることができる。

また、学習データの前意図情報を用いることで、式 (2) を以下の式 (3) のように書き換えることができる。

$$\hat{I} = \underset{I}{argmax}\, P(I^u|I^{u-1})\prod^{n} P(w_n|I^u, w_{n-1}, \cdots, n_1) \quad (3)$$

ここで、I_u とは、ある発話 u における意図のことであり、I^{u-1} は、発話 u の一つ前に出現した発話の意図である。よって、確率 $P(I_u|I_{u-1})$ を求め、式に組み込むことで、前意図情報を用いた意図推定を行うことができる。

N-gram モデルの作成を行う場合、単語列の出現頻度から確率値を推定するが、ゼロ頻度問題を解決するために、頻度情報のディスカウンティングを行う。ディスカウンティングには、リスタッド法 [8] を用いた。

3.2 ベクトル空間モデル

ベクトル空間モデルは、文書検索の分野でしばしば用いられる手法である。文書をキーワードによるベクトルモデルで表現し、質問から生成したベクトルとの類似度を求めることで、該当文書を検索する。ベクトル空間モデルを用いた意図推定としては、各意図を 1 つのベクトルで表現する方法と、各意図を複数の代表ベクトルで表現する方法の 2 つを評価した。それぞれの具体的な内容を以下に示す。

3.2.1 各意図を 1 つのベクトルで表現する方法

この方法は、文書検索に用いられるベクトル空間モデルにおいて、文書にあたる部分を意図に置き換えて行った方法である。

この方法におけるモデル作成は以下の手順で行う。

1. 全発話を意図毎に分類する
2. 各意図における単語 (列) の出現頻度を数える
3. 単語 (列) の出現頻度をもとに、各意図のベクトルを tf/idf 法により求める

tf/idf 法とは、重み付けを行ったベクトル生成手法の一つである。tf(term frequency) とは、ターム頻度のことであり、式 (4) に示すように頻度をタームの出現数で割った相対頻度を用いた。

また、意図推定用のベクトルモデルとするため、文書にあたる部分を意図と置き換えた。つまり、tf_{it} は意図 i におけるターム t の重みを表す。

$$tf_{it} = \frac{freq(t, i)}{\text{意図 } i \text{ 中の出現ターム数}} \quad (4)$$

idf(inverse document frequency) は、タームが全文書中のどれくらいの文書に出現するかを表す尺度である。idf を式 (5) に示す。ここでも、意図推定用のベクトルモデルとするため、文書頻度を意図頻度と置き換えた。つまり、df_t はターム t が出現する意図数である。

$$idf_t = \log \frac{N}{df_t} + 1 \quad (5)$$

意図 i におけるターム t の重み w_{it} は式 (6) であり、

$$w_{it} = tf_{it} \times idf_t \quad (6)$$

これにより、意図 I_i のベクトルは式 (7) で求まる。

$$I_i = (w_{i1}, w_{i2}, \ldots w_{it}, \ldots, w_{in}) \quad (7)$$

3.2.2 各意図を複数の代表ベクトルで表現する方法

各意図を 1 つだけではなく、複数のベクトルで表現する方法である。つまり、事例をそのままベクトル表現にし、似たベクトルを統合していくことで、各意図の代表となる複数のベクトルを決める。

この方法におけるモデル作成は以下の手順で行う。

1. 各発話における単語 (列) の出現頻度を数える
2. 単語 (列) の出現頻度をもとに、各発話のベクトルを tf/idf 法により求める
3. 意図毎に各データを分類する
4. 分類された意図内において、ある 2 つのデータ間の類似度を計算する
5. 類似度が閾値を超えた場合、その 2 つのデータの平均ベクトルを求めて、1 つにまとめる
6. 閾値を超えるデータがなくなるまで 4、5 を繰り返す

tf/idf 法は、3.2.1 節のものとほぼ同じであるが、以下の式 (8) に示すように意図 i にあたる部分は、発話を統合した意図 i のサブグループ g として扱った。

$$tf_{gt} = \frac{freq(t, g)}{\text{サブグループ } g \text{ 中の出現ターム数}} \quad (8)$$

3.3 Support Vector Machine (SVM)

SVM[5] は、訓練サンプルからの学習により分離超平面を求め、それにより 2 クラスのパターン認識を行うモデルである。しかし、一般にこの分離超平面は一意に決まらない。そこで、未学習のデータに対してもうまく分離できるように「マージン最大化」という基準を用いて、この分離超平面を求めている。訓練サンプル集合 (各データは (x_1, x_2) の 2 次元データ) が 2 つのクラスに線形分離可能な場合の分離超平面 (ここでは直線) を図 3 に示す。

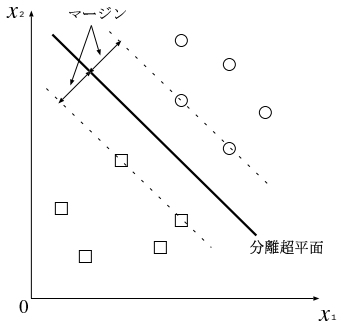

図 3: 分離超平面とマージン

上記の SVM は、訓練サンプルが線形分離可能な場合の例であるが、実問題で線形分離可能な場合は稀である。そこで、多少の識別誤りは許す「ソフトマージン」と呼ばれる方法を用いて、この問題を解決している。しかし、ソフトマージン法を用いたとしても、非線形で複雑な問題には、良い識別を行うことができない。そこで、本質的に非線形な問題に対応するため「カーネル学習法」と呼ばれる方法が用いられている。これは、訓練サンプルの特徴ベクトルを非線形変換し、その空間で線形の識別を行う方法である。SVM は、これらの方法を利用することで、認識精度の高い識別器を構成できる学習モデルである。

4 対話コーパス

収集した対話データ、学習データの作成手順を以下に示す。対話は、雑談のようなものではなく、ある目的を達成するための対話、つまりタスク指向対話を想定している。本研究では「ホテル検索・予約」タスクのもとで、対話を収集した。

4.1 対話収集環境とコーパスの情報

収集したコーパスに「ホテル検索・予約」タスクに関する対話である。対話はゲスト役とオペレータ役の 2 人で行い、ジェスチャーなど音声情報以外の伝達はできないように非対面の環境下で行った。ゲストに、発話の制約は与えなかったが、予め用意したシナリオの検索条件にできるだけ沿うように対話を進めてもらった。収集した対話に関して表 2 に示す。

表 2: 対話コーパス

対話データ数		100 対話
被験者数		23 人
異なり単語数		797 単語
発話数	ゲスト	2806 発話
	オペレータ	3226 発話

4.2 学習データの作成

学習データの作成手順を以下に示す。作業 4、5 に関しては手作業により行った。

1. 収集した音声対話データを書き起こす
2. 発話の一般化のため、固有名詞・数値をクラスラベルに置き換える
3. 茶筌 [7] を用いて、形態素解析し、形態素列データを生成する
4. 発話を意図単位で分割
5. 分割した全発話に正解とする意図を付与

4.3 クラスラベル

発話中に出現する特定の名称･数値は、クラスラベルに置き換えることによって発話の一般化を図った。

収集対話に施したクラスラベル変換の例を表 3 に示す。また、最終的に作成された意図タグ付き対話コーパスの例を表 4 に示す。

表 3: クラスラベルへの変換

出現単語		＜クラスラベル＞
N 月,N 日	→	＜宿泊日＞
場所, 地名	→	＜地名＞
N 円	→	＜料金＞

注 : N は数値

表 4: 発話意図を付与した対話データ

発話	発話意図
はい、ホテルの予約システムです。	[対話開始]
えー、＜宿泊日＞に (はい)	[宿泊日・期間提示]
＜地名＞の (はい) ＜地名＞付近の	[場所提示]
ホテルを探しているんですが。	[ホテル検索依頼]
＜地名＞付近。	[場所確認]
え、ご宿泊は＜宿泊日＞でよろしいですか	[宿泊日・期間確認]
はい、できれば (はい) 移動を＜時間＞以内のところがいいんですけど。	[場所提示]

() 内は相槌

5 評価実験

3 章で述べたそれぞれの手法の意図理解精度を評価するため、以下のような実験を行った。収集データにはゲスト (ユーザ) の発話だけでなく、オペレータ (システム) の発話も含まれているが、混在させて学習させたものよりも各発話者毎にモデルを作成した方が精度が良かったため、以下の実験では、これらを分離しユーザ発話のモデルのみ評価を行った。また、各実験で用いた前意図情報とは、学習データに含まれる前意図データから、前意図毎の現在の意図の確率を求めたものである。また、ユーザ発話における評価データ内の出現意図は 48 種類であり、出現頻度は 1 のものから数百のものと偏りがある。

5.1 N-gram モデルの評価実験

N-gram モデルにおける、意図推定精度の実験を行った。実際には、uni-gram、bi-gram、tri-gram においてモデルの作成・評価を行った。実験は、前意図情報の有効性を示すため、この情報を使用した場合とそうでない場合の両方を行った。モデルの評価は学習データを 10 のグループに分け、leave-one-out 法を用いて行った。実験結果は 10 回の評価結果の平均値である。

各モデルの意図推定精度の結果を表 5 に示す。また、前意図情報だけで意図推定を行ったところ、49.9%の正解率であった。

表 5: N-gram モデルの意図正解率 :単位 (%)

前意図	uni-gram		bi-gram		tri-gram	
	closed	open	closed	open	closed	open
使用	93.8	74.8	95.6	75.3	95.6	72.7
未使用	85.6	66.8	94.8	70.5	96.0	69.1

結果から、前意図情報を用いた bi-gram が最も良い結果となった。tri-gram の正解率が下がったのは、学習データの不足が原因と考えられる。前意図情報を用いたものとそうでないものを比較すると、前意図情報を用いたほうが全てのモデルに関して良くなっており、また、前意図情報だけでも 50%の意図推定ができることから、この情報はかなり有効であることが分かる。

5.2 ベクトル空間モデルの評価実験

ベクトル空間モデルにおける意図推定は、推定対象のベクトルと各意図のベクトルとの類似度をコサイン相関値を用いて計算し、類似度が最大となる意図を、推定意図とした。

ただし、最大類似度となるデータが複数存在する場合は、前意図情報 (確率値) を用いて比較することで、より尤もらしいものを選択するようにした。(その値も、同値である場合は誤りとした)

表 6: ベクトル空間モデルの意図正解率 :単位 (%)
(各意図を 1 つのベクトルで表現した方法)

uni-gram		bi-gram	
closed	open	closed	open
51.9	43.3	65.3	55.3

表 7: ベクトル空間モデルの意図正解率 :単位 (%)
(複数の代表ベクトルで表現した方法 :閾値 1)

uni-gram		bi-gram	
closed	open	closed	open
86.9	66.7	90.8	68.2

また、タームを単語として扱うか、単語連鎖として扱うかで、生成するベクトル値が変わってくる。今回は、単語 (uni-gram)、単語連鎖 (bi-gram) の 2 種類を生成し、それぞれを評価した。

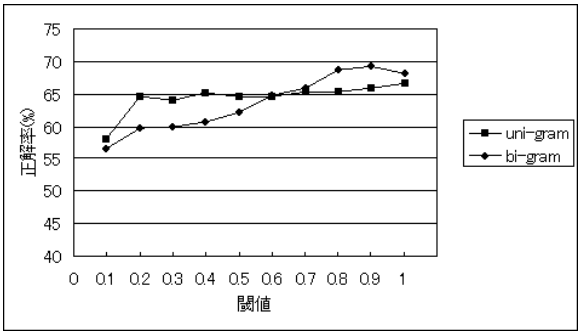

図 4: 閾値と意図正解率

3.2.1 節の各意図を 1 つのベクトルで表現した方法の評価結果を表 6 に示す。3.2.2 節の各意図を複数の代表ベクトルで表現する方法においては、閾値を 0.1 から 1 まで 0.1 毎変化させながら評価実験を行った。閾値によって統合した後のベクトル数としては、閾値 0.3 で 939、閾値 0.6 で 1568、閾値 0.9 で 2195 となった。

結果を図 4 に示す。また、2 つの方法の比較のため閾値 1 における意図正解率を表 7 に示す。

各意図を 1 つのベクトルで表現した場合、N-gram モデルと比べて推定精度がかなり低い。これは、各発話の単語情報をあまりにも抽象化しすぎてしまったのが原因と考えられる。また、最大類似度を示すデータが複数存在する場合は全体の 4 割程度と多く、前意図情報が有効に働いていると考えられる。各意図を複数の代表ベクトルで表現する場合は、前者の方法よりは精度が良かったが、より良い精度を得るには、ベクトルデータをできるだけ統合せずにそのまま残す必要がある。しかし、これだと類似度計算における計算量が膨大になるという問題が発生する。ベクトル空間モデルにおける手法はさらなる検討が必要である。

表 8: SVM の意図正解率 :単位 (%)

kernel	uni-gram		bi-gram	
	closed	open	closed	open
linear	90.0	66.5	91.0	69.2
gaussian	90.3	67.2	92.5	70.2

5.3 SVM の評価実験

今回、SVM の作成・評価には、SVMTorchII[9] というツールを用いた。このツールは one-per-class という方法により複数クラスの分類が可能である。これは、各クラス毎に判別器を生成し、判別器の返す値 (kernel による) の中で最大値を示す判別器のクラスを推定クラスとする方法である。また、カーネルの選択も可能である。今回の実験は、linear kernel と gaussian kernel の 2 つの kernel で行った。

学習するデータの形式は、タームの出現頻度によるベクトルデータを用いた。そのため、ベクトル空間モデル同様、uni-gram 型と bi-gram 型の 2 種類の学習データを用意した。ただし、今回はツールの特性上、前意図情報を利用せずに意図推定実験を行った。SVM を用いた手法の実験結果を表 8 に示す。

SVM では、上記の 2 つの方法 (N-gram モデル、ベクトル空間モデル) と違い、前発話の意図情報を用いていない。つまり、発話の表層的な情報のみで意図推定を行っているため、期待した程の結果は得られなかった。しかし、前意図情報を用いずに約 70%の意図推定が行えていることから、前意図情報をうまく活用することで、更なる精度の向上が期待できる。

5.4 学習データ量による推定精度の実験

学習データ量の変化によって、推定精度がどのように変化するか、また、手法によって最低限必要な学習データ量に違いがあるのかを評価するため、学習データ量を変化させた意図推定実験を行った。

10 グループあるデータにおいて、評価データをある 1 グループで固定し、学習用データの数を変化させることで、推定精度の違いを比べた。

実験を行ったモデルは、前意図情報を用いた uni-gram モデル、bi-gram モデルと、ベクトル空間モデル (各意図を複数の代表ベクトルで表現する方法データ形式が bi-gram のもの) である。結果を図 5 に示す。

どのモデルにおいても、学習データ数が 1000 未満だとデータ数の増加で精度が大きく変化する。その後もデータ数を増加させると精度は向上するが、その変化は決して大きくない。このことから、ある程度安定した推定結果を得るには、用意する学習データ数として 1000 以上にすることが一つの基準と言える。結果として bi-gram モデルの推定結果がほぼどのデータ数でも最も良い値を示しているが、推定精度の変化は他のモデルも同じような傾向を見せている。よって、データ数の変化によってモデル毎の大きな違いはなかった。

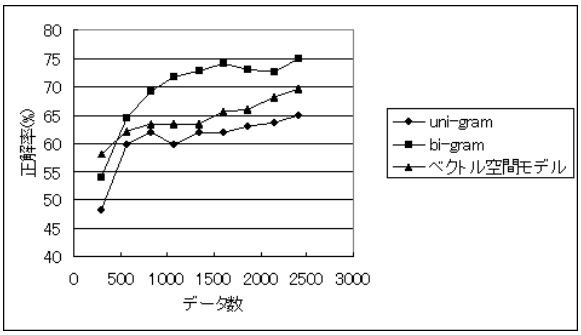

図 5: データ量による推定精度の変化

6 おわりに

本論文では、自然発話の意味理解に発話意図の推定を用いた手法を提案した。意図推定モデルの作成には、N-gram モデル、ベクトル空間モデル、SVM を用い、各手法の意図推定精度を評価した。各モデルの評価結果から、前意図情報が非常に有効であることが分かった。また、データ数の変化による推定精度の違いを評価してみたところ、モデルによる違いはほとんどなく、用意する学習データの基準として 1000 以上は必要であることが分かった。最も良い結果のものは、前意図情報を用いた bi-gram モデルで 75.3%の正解率であったが、意図推定精度がこの程度で十分かどうかは、現段階ではまだ言及できないため、実際に対話システムを構築した上で評価を行う必要がある。今後の課題としては、複数意図を持つ発話の分割手法の提案、音声認識結果からの統計的な意図理解の精度評価、本手法を用いた対話システムの構築が挙げられる。

参考文献

[1] Victor Zue, Stephanie Seneff, James Glass, Joseph Polifroni, Chiristine Pao, Timothy J. Hazen and Lee Hetherington, "JUPITER: A telephone-based conversational interface for weather information," IEEE Transaction on Speech and Audio Processing, Vol.8, No.1, January 2000.

[2] Diane Litman, Shimei Pan and Marilyn Walker, "Evaluating Response Strategies in a Web-Based Spoken Dialogue Agent." In Proceeding of ACL/COLING 98, 1998.

[3] 北研二: 言語と計算 4 確率的言語モデル, 東京大学出版会, 1999.

[4] 徳永健伸: 言語と計算 5 情報検索と言語処理, 東京大学出版会, 1999.

[5] Burger, C.J.C., "A Tutorial on Support Vector Machines for Pattern Recognition" in Data Mining and Knowledge Discovery, 1998. 2(2):pp.121-167.

[6] 荒木雅弘, 伊藤敏彦, 熊谷智子, 石崎雅人 "発話単位タグ標準化案の作成," 人工知能学会誌 Vol.14, No.2, pp.253-255, Mar. 1999.

[7] http://chasen.aist-nara.ac.jp/

[8] Ristad, E.S. "A Natural Law of Succession," Research report CS-TR-495-95, Princeton University, July 1995.

[9] http://www.idiap.ch/learning/SVMTorch.html