

# 自然言語コーパスにおける「笑い」を利用したソフトウェアの要求獲得と情報検索への応用

土井 晃一\*

\*株式会社 富士通研究所 マルチメディアシステム研究所

〒261 千葉県千葉市美浜区中瀬1丁目9番地3号(幕張システムラボラトリ)

(043)299-3100

doy@iias.flab.fujitsu.co.jp

## 1 はじめに

古来、笑いに関する研究は多い。しかし、笑いのあまりの複雑さにその全貌が解明されているとは言い難い。

Bergson[1]によると、ある種の笑いは「一つの特殊社会の習俗なり観念なりと相関的」であり、「職業のおかしみ」が存在するとしている。また、森下[2]は、それまでの笑いの研究をまとめた。それによると、ある種の笑いは、道徳的なもの・トリックスター・様々な規範と関係する。

つまり、笑いを分析することにより、職場風土、本音、極論を知ることができる可能性がある。そこで本稿では、笑いの分析をソフトウェアの要求獲得と情報検索に利用する方法について述べる。

## 2 コーパスの有用性

本節では、コーパスを用いた談話分析の有用性について述べる。

言語コーパスは作るまでが大変である。話し言葉を書き起こした言語コーパスの場合、どれぐらいの品質のものを作るかにも大きく依存するが、目安としては、10分書き起こすのに、約1時間かかる。他人に頼むにしても、人間がやる仕事であるから、依頼人が望んだような品質のものが出来上がるとは限らない。その結果、往々にして、後からかなり手を入れることになり、初めから自分で作った方が良かったと思うこともしばしばである。

しかし、一度コーパスが出来てしまえば、当然のことながら、便利なることはいろいろとある。これは、実際の発話を聞きながら分析を進める発話ベースや、発話を録音したものをもとに分析を進めるテープベースに比べるとその有意性は明らかである。

まず、コーパスは一覧性に優れている。テープベースの場合だと、問題にしようとしている言語現象の前後を見るのは困難であり、えてして非常に面倒な動作が伴うことが多い。さらに、発話ベースの場合には不可能である。これらは、分析に多大の障害をもたらすことが多い。

また、コーパスは発話ベースやテープベースに比べるとゆっくり分析することが出来る。テープベースでさえ、分析に必要な言語現象を聞き逃してしまうことがあり、また、かなりの記憶力が必要となる。また、発話ベースでは、考えているうちに次に進んでしまうこともままある。

最後に、コーパスは言語現象の位置の特定に優れている。発話ベースやテープベースに比べると、ある言語現象が実際に発話のどの場所に現れたかを正確に特定することが出来る。

もっとも、コーパスはあくまでも「書き起こした」ものである。イントネーションやニュアンスやアクセントなどを含んだ臨場感が欠如しているため、思わぬ勘違いを生じさせることもある。イントネーションやアクセントなどを記述する方法も存在するが、書き起こし・読みとり共に使いこなすには相当な技術が要求される。

## 3 笑いと規範

本節では、笑いと規範との関係を実例とデータを元に議論する。

言語行為に引き続いて発生する笑いの中には、何らかの意味で社会的な規範と関係するものがある[2]。笑いとは社会的な規範との関係は、規範を破ることであったり、規範とのズレが想起されるものであったり、単に規範が想起されるだけであったりする。

規範には、言語規範(例えば文法)、思考規範(例えば排中律、同一律)、行為規範(例えば道徳、規則、法律)がある。例えば、言語規範と関係する笑いには、駄洒落・地口・誇張などが挙げられる。

実際のソフトウェアの要求獲得会議中に現れた例を元にして説明する。まず、言語規範だけに関係する笑いの例としては、以下の例がある。

「それはそれなりにターミナル認識して臨機応変(笑)にやってもらえると、嬉しいんですけど。」

(笑)がある箇所が、笑いの起きた場所である。この(笑)は笑いが起きたことを意味しているだけで誰が笑ったかは示していない。つまり、この場合必ずしも発話者が笑ったことを意味していない。この発話は、画面の表示がキャラクタ端末・X端末それぞれに「適切に」表示して欲しいことを述べた発話である。「適切に」という意味を「臨機応変」という表現で誇張しているところが言語規範と関係するところである。

次に思考規範だけに関係する笑いの例としては、以下の例がある。

「だって、もともと一人の、一人の人は二つの会議に出席しないんだから、いまのところは。(笑)」

この発話は、一人の人は二つの会議に同時に出席することが「論理的に」ありえないことを指摘した発話である。この「論理的に」ありえないところが思考規範と関係している。また、逆に論理的に当然のことを言って笑いを誘う場合もこの範疇に入る。

最後に行為規範だけに関係する笑いの例としては、以下の例がある。

「で、いないときのメモの置き場にもちょっと悩むときがあって(笑)、だからうーん。」

この発話は、書記が電話の伝言をメモで研究員に渡そうとするとき、机の上が散らかっていて、わかりやすい場所にメモを置かないことを指摘した発話である。机の上は整理してあるべきであるという行為規範に関係している。

ここでは説明のために単一の規範だけに関係する笑いの例を示したが、実際には二つあるいは三つすべての規範に関係する笑いも数多く存在する。

会議の中に発生した笑いが、言語規範・思考規範・行為規範のいずれと関係しているかを実際に調べてみると以下の表1のようになる。

	会議1	会議2	会議3
言語規範	5	8	1
思考規範	26	26	11
行為規範	33	26	17
その他	4	3	4
笑いの総数	56	44	26

表1: 笑いの性質

対象とした会議はいずれもソフトウェアの要求獲得会議であり、会議の所要時間はそれぞれ約3時間である。会議はそれぞれ書き起こしを行ない、笑いの場所を明記した上で、笑いの直前の発話と規範との関係を調べた。表中の数字はそれぞれの規範に当てはまる笑いの数である。一つの笑いが他数の規範を含む場合があるので、縦の欄の合計は笑いの総数にはならない。その中の中には、分析不能なものも含まれている。

この表からわかることは、まず第一に言語規範と関係のある笑いの比率が低い(会議1で9%、会議2で18%、会議3で4%)ことである。ソフトウェアの要求獲得会議の場は一応改まった場であるので、駄洒落・地口のような一般にはくだけた表現と思われるものは出現しにくいものと思われる。

次に行為規範と関係のある笑いの比率が高い(会議1で59%、会議2で59%、会議3で65%)ことである。

すべてがそうであるとは言えないが、何らかの職場のルール(すなわち職場風土)と関係しているものが多いことがわかる。

## 4 要求獲得オフライン法での利用

本節では、ソフトウェアの要求獲得で関心事項を抽出するための実現法の一つとして笑いをいかに利用するかを述べる。

ソフトウェア要求獲得プロセスでよく使われている方法として会議が挙げられる。会議では「要求獲得者の容認と理解」というフィルター、時間的な制約などにより、要求獲

得が的確に行われないことがある。つまり要求を網羅的にキャッチアップするのが困難であり、話題の展開が不十分なまま終わってしまうこともあり得る。これらの問題点を解決するために、会議をビデオ撮影して、オフラインで観察・分析する方法(オフライン法)がある[3]。

また、ソフトウェア開発に当たっては、下流ではコストをかけるが上流ではあまりコストをかけない。そのため、顧客の本当の要求が正確に獲得できないで、システム構築に入ってしまうことがよくある。その結果、使いにくい、あるいは、使われないシステムが往々にして出来上がる。使いにくい、あるいは、使われないシステムを作らないために、上流、特に要求獲得のフェーズではかなりのコストをかけてもよい、と考えられる[4]。

以上のような問題意識を背景にして、我々はオフライン法の一実現方法として USP-Offline 法を構築している[5]。USP-Offline 法では、ビデオテープから発話を書き起こし、コーパスを作ることから解析を始める。USP-Offline の主な出力としては、構造化された精密な議事録と関心事項があるが、本稿ではその関心事項側について述べる。

関心事項の抽出の仕方の一例として、ここでは笑いを取り上げる。笑いのオフライン法への利用は以下の三点が考えられる。

1. 笑いの内容がわかったかどうかを確認する
2. 笑いの本音を推論する
3. 笑いの極論を推論する

まず1について考察してみる。ここで言う「わかった」は、書き起こしの時に笑いがあったことをわかったかどうかではなく、その笑いがわかったかどうかを指す。会議中の笑いが分析者に理解できるかどうか(必ずしも一緒に笑って笑う必要はない)は、分析者が会議をどれくらい理解したかに依存する。つまり会議の背景にある、会議参加者の職場風土・会議参加者の性格や興味・会議中に現れた概念を分析者がいかに理解しているかに大きく依存する。

分析者が会議中の笑いが理解できないまま放置しておく、と、会議参加者の職場風土・会議参加者の性格や興味・会議中に出てくる概念を理解しないまま、最悪の場合は誤解したまま、分析を行なうことになり、正確な分析が得られない。

会議中に現れた笑いの中で分析者に理解できない笑いが分析中に現れた場合、その笑いの意味(なぜおかしいのか)を分析者は調べる必要があるものと考えられる。さらに一応理解できた笑いも念のため会議参加者に確認する必要があると考えられる。厳密に言えば、誰が笑ったか(発話者だけが/発話者以外の全員が/会議参加者の全員が)を知ることができるが、ここでは、まず研究の第一段階として、笑いが起きたかどうかだけを対象にする。

先ほどの例をもう一度挙げて説明する。

「で、いないときのメモの置き場にもちょっと悩むときがあって(笑)、だからうーん。」

この場合、前述の通り机の上が散らかっていることが笑いの背景にあることが、実際に発話されていなくても推論できる。分析者がこの笑いを理解できたということは、この背景が理解できたということ意味する。

次に2について考察してみる。笑いが特に行為規範に関係している場合、その行為規範はその会議の参加者の職場風土の行為規範であることがある。この行為規範は笑いの本音を推論することによって得られることがある。つまり、笑いの本音を推論することによって、会議の発言の中には現れにくく、かつ第三者にはわかりにくい会議参加者の職場風土の行為規範を知ることができる場合がある。

また、笑いの本音を推論することによって、要求項目の内容をより精密に判断・作成することもできる。

例を挙げて説明する。

「そうすと、ゴミ、ゴミの山のなかじゃなくて、  
メールの山のなかで、結局、一緒かも知んない  
んだけど、実は(笑)。」

この発話は、電話の伝言を紙のメモではなく、電子メールで行なうことを提案したもののに対して、結局情報量が多過ぎてわからなくなること指摘した発話である。例えば、電子メールの来る量が多いことが問題であることがこの笑いから見てとれる。しかも、伝言を電子化したところで問題の解決にはつながらないと言話者が判断しているのではないかと分析者は推論できる。

最後に3について考察してみる。笑いの中には考えられる解決策の中でもっとも極端な解決策を示された時の反応であることがある。この場合、解決策の範囲が示されたことになる。もし解決策が存在するならば、このもっとも極端な解決策の内側に存在することになる。

これも例を挙げて説明する。

「出張だったとすれば、それは勝手に WINS に  
入って、全部精算までしてくれるというのが、  
理想なの。(笑)」

この発話は、スケジュール管理のプログラムに入力しておけば、出張の旅費清算もすべて自動的にやってくれることを要求した発話である。ここではいくつかの解決のレベルが考えられる。

1. 出張の旅費清算はスケジュール管理のプログラムと完全に独立に行なう
2. 出張の旅費清算はスケジュール管理のプログラムを参考にして行なう
3. 出張の旅費清算はスケジュール管理のプログラムから自動的に行なう

このように笑いのひき起こされた発話では一番極端な解決策が示されたことになる。

## 5 笑いを利用した関心事項の抽出

本節では、笑いを利用した関心事項の抽出法について述べる。二つの方法が考えられる。一つの方法は会議中に笑いを調べる専門のスタッフがいて、笑いの起きたところを忠実に記録する方法である。もう一つの方法は発話録(コーパス)を起こし、笑いの起きたところを正確に把握する方法である。この二つの方法は正確さと工数がトレードオフの関係になっている。

いずれの方法をとるにせよ、会議中に笑いの起きた場所とその文脈が記述できる。次に、分析者はその笑いの起きた場所と文脈を見ながら、

1. その笑いの内容がわかったかどうか?
2. その笑いの本音は何か?
3. その笑いが示している極論は何か?
4. その笑いが含意している職場風土は何か?

を推論する。1は基本的にわかったかわからなかったかだけの情報を持つ。2,3,4は推論事項となる。当然のことながら、その笑いがわかってもらただ笑わせるだけの笑いのように、役に立たないものは捨てる。本音・極論も役に立たないものは

捨てる。注意しなければならないのは、役に立つ・立たない基準は当該システムに関係があるかどうかではない。その笑いが一般的かどうか、言い換えると、その笑いがその社会の外でも理解できるかどうかである(必ずしも一緒になって笑う必要はない)。その社会の外でも理解できる笑いは役に立たない笑いであり、その社会の中でしか理解できない笑いが役に立つ笑いである。これはその社会の中でしか理解できない笑いがその社会の風土・習慣などを背景にしていることがあるからである。ここで注意しなければならないのは、必ずしも分析は完全である必要はない。この推論結果は次の段階のトリガに過ぎないからである。

分析者はその結果を元にコーディネータとともに次の会議の前に事前会議をする。事前会議では、1,2,3,4についてお互いに話し合い、次の本会議で話し合う事項を決める。この事項はどれぐらい笑いが理解できないかが選出の基準になる。理解できない笑いほど優先順位が高い。前述のように、その社会でしか通用しないの中には、その社会独特の風土・習慣を背景にしていることがあるからである。ただ、これらの事項は、システムの構築に携わる開発側にも知らせられない性格のものである可能性があるため、扱いには注意しなければならない。本会議でどうも話題展開がすすきりいかないときは、開発者・コーディネータが席を外して、顧客・ユーザだけの会議で解決をはかってもらうことも必要となろう。いずれにしてもコーディネータの裁量が要求される。

## 6 試行

本節ではこの関心事項の抽出の方法論を部分的に適用した結果について論ずる。

コーパスから笑いを抽出して、その笑いの内容がわかったかどうか・その笑いの本音は何か・その笑いが示している極論は何かを実際に推論してみた。推論結果は確かに職場風土を如実に表しているものが多く、この場で発表できないものも多い。このことがまさにこの方法論の良さを示しているともいえる。残念ではあるが、当たり触りのない例を挙げて説明することにする。ここでは説明のため、理解可能でかつ本音も極論も容易に推論できる例を挙げる。前述の例をもう一度使う。

「出張だったとすれば、それは勝手に WINS に  
入って、全部精算までしてくれるというのが、  
理想なの。(笑)」

この例の解析結果は、

理解 ○

本音 無理だろうけど

極論 全部自動的にやる

職場風土 事務処理をいやがる

のようになる。まずこの笑い内容の理解はできたので、理解の欄は○になる。次に本音の欄は、発話者が実際には無理だろうと思っていることが推論されるので、「無理だろうけど」となる。また、極論の欄は、発話者が全部自動的にやって欲しいことが推論されるので、「全部自動的にやる」となる。最後に、職場風土の欄は、事務処理をいやがっていることが推論されるので、「事務処理をいやがる」となる。

分析をしていて気づいたことは、本音と極論は概念の上ではまったく別のことであるが、実際に分析してみると、分離しにくかったことである。ものよっては二つの欄に同じことを書かなければならないことも生じた。これは本音と極論

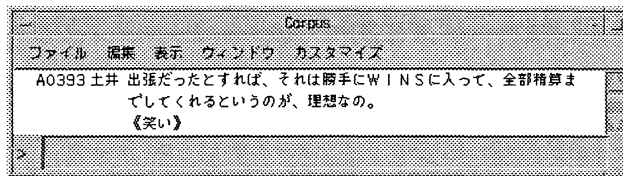


図 1: コーパスウインドウの例

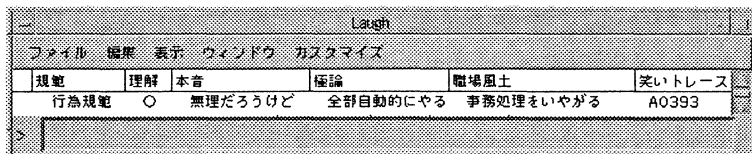


図 2: 笑いウインドウの例

が必ずしも直交している概念ではないためであると考えられる。

また、これらの会議中に現れたすべての例は、直前の発話が笑いの直接の原因になっているものであった。「形・態度・身振り・状況・行動」が笑いの原因になったものはいなかった。

コーディネータと分析者による事前会議、並びに本会議へのフィードバックは残念ながらできなかった。

## 7 笑い分析のためのケースツール

本節では、前述までの方法論を効率的に行なうためのケースツールについて述べる。

本ケースツールは、USP-Offline 法のケースツールとして開発されたものである。本稿では、笑い分析に関連するところだけについて述べる。

笑い分析のケースツールは、コーパス作成プロセスと関心事項分析プロセスの笑いウインドウからなる。コーパス作成プロセスはコーパスウインドウを用いて行なう。図 1 にコーパスウインドウの例を示す。分析者はこのコーパスウインドウを用いて、書き起こしを行なう。このコーパス作成が終わったら次に図 2 に示すような笑いウインドウを用いて笑いの分析を行なう。笑いの部分はプログラムにより自動的に抽出され、そのトレースは自動的に張られる。こうすることによって分析後の追跡が容易になる。

## 8 情報検索での笑いの利用

本節では、情報検索における笑い情報の利用について議論する。

前述のように、笑いの中には何らかの本音が含まれていることが多い。これを例えばネットワークニュースというドメインで考えてみると、笑いの数をグループ毎、あるいは個人毎に累加して統計をとることによって、本音が話されている可能性が高いグループ/個人などが推定できる可能性がある

る。このように情報検索に心理的軸を持ち込むことによって、情報検索に新たな展開が考えられる。さらに、ネットワーク社会学のようなものも可能になるであろう。

## 9 おわりに

本稿では、ソフトウェアの要求獲得会議から会議参加者の本音・極論・職場風土を導き出すための方法論として笑いをとりあげ、その考え方・方法論・試行・ケースツール化について述べた。また、情報検索での利用の仕方について議論した。今後は、要求獲得の場合は、実際の会議へのフィードバックをしてみてその有用性について確かめる予定である。また、情報検索については、システムを構築して、その効果のほどを調べる予定である。

## 参考文献

- [1] ベルグソン, 林達夫訳. 笑い. 岩波書店, 1986.
- [2] 森下伸也. ユーモアの社会学. 世界思想社, 1996.
- [3] 大森晃, 土井晃一. オフライン要求獲得法の提案. 情報処理学会第 48 回全国大会, Vol. 5, No. 4K-6, pp. 373-374, 3 月 1994.
- [4] 大森晃, 土井晃一. 要求獲得法に関する提案. ソフトウェアの要求分析に関するシンポジウム, pp. 35-48, 10 月 1993.
- [5] 片山佳則, 蓬萊尚幸, 渡部勇, 土井晃一, 園部正幸. ユーザ指向ソフトウェア開発のための要求分析法の実践について. 日本ソフトウェア科学会 ソフトウェアプロセス研究会, 3 月 1996.