平成 22 年度 AAMT/Japio 特許翻訳研究会

報　告　書

機械翻訳及び辞書構築に関する研究
及び
シンポジウム報告

平成 23 年 3 月

一般財団法人 日本特許情報機構

目　　次

AAMT／Japio 特許翻訳研究会委員名簿

（敬称略・順不同）

委 員 長	辻井　潤一	東京大学大学院教授・マンチェスター大学教授 AAMT 前会長
副委員長	横山　晶一	山形大学大学院教授
〃	江原　暉将	山梨英和大学教授
委　　員	宮澤　信一郎	秀明大学教授
〃	梶　　博行	静岡大学教授
〃	黒橋　禎夫	京都大学大学院教授
〃	宇津呂　武仁	筑波大学大学院准教授
〃	二宮　崇	愛媛大学大学院准教授
〃	越前谷　博	北海学園大学助教
〃	綱川　隆司	静岡大学学術研究員
〃	範　　暁蓉	東京大学大学院　中川研究室
〃	王　　向莉	東京大学大学院　辻井研究室　特任研究員
〃	安田　圭志	(独)情報通信研究機構
〃	熊野　明	東芝ソリューション(株)
〃	下畑　さより	沖電気工業(株)
〃	潮田　明	(株)富士通研究所
〃	三浦　貢	日本電気(株)
事 務 局	村上　嘉陽	AAMT/Japio 特許翻訳研究会東京事務局・(株)ナビックス
〃	河田　容英	〃　〃　〃
〃	高田　佳代子	〃　〃
オブザーバー	中川　裕志	東京大学大学院教授
〃	安藤　進	元多摩美術大学講師
〃	守屋　敏道	(財)日本特許情報機構
〃	森藤　淳志	〃
〃	藤城　享	〃
〃	大塩　只明	〃
〃	塙　　金治	〃
〃	三橋　朋晴	〃
〃	柿田　剛史	〃
〃	土屋　雅史	〃
〃	星山　直人	〃

１．　は　じ　め　に

東京大学大学院情報理工学系研究科　教授
マンチェスター大学コンピュータ科学科　教授(兼任)
AAMT/Japio 特許翻訳研究会委員長
辻井　潤一

AAMT（アジア太平洋機械翻訳協会）の特許翻訳研究会は、Japio（日本特許情報機構）からの委託をうけ、本年度も 7 回にわたる委員会を開催し、中国・韓国の研究開発グループとの連携を強化するなど、活発な研究調査活動を行ってきた。本報告書は、2010 年度の本委員会での活動と各委員の研究成果を報告するものである。

本委員会が始まった平成 15 年以降、年ごとに特許情報の有効活用、その機械翻訳に対する言語処理技術への関心が急速に高まってきている。特に、中国・韓国からの特許出願数が急増するなど、世界の中でのアジア圏の存在感が高まってきている。本年度は、その重要性を増しつつあるアジア言語間の特許翻訳についての調査活動にも精力を注ぎ、中韓 2 国からの招待講演を含むシンポジウム（参加者約 150 名）を開催するなど、活発な活動を行った。中韓 2 国の研究グループとは、今後も交流を活発化することで合意し、2011 年度にアジアで開催予定のＭＴサミットに合わせて特許翻訳に焦点を当てたワークショップを本委員会が中心となって計画している。

ヨーロッパの翻訳局でＧｏｏｇｌｅによる翻訳サービスを活用することが本格的に検討されるなど、特許文書への機械翻訳の適用への関心は急速に高まってきている。ただ、一方では、日本語・中国語・韓国語などのアジア言語の機械翻訳システムの性能は、ヨーロッパ語族間のそれに比べると困難とされ、実際、翻訳精度の点でヨーロッパ語族間の翻訳システムの性能よりもはるかに低いものである。この現状を改善するためには、本委員会内の活動だけでなく、中国語・日本語･韓国語の処理のための基礎技術を研究するグループとの共同も重要である。本委員会では、このような認識から、言語処理の基礎的な技術研究とその特許翻訳への適用にも力を注いでいる。本報告書では、その研究成果をまとめたものとなっている。

本委員会が主催する発表会やシンポジウムへの参加者は、年とともに増加してきている。本年度はこれまでにも増して特許情報への関心、とくにアジア諸国での特許情報の交流への関心が高まった年であった。この関心の高まりを反映して、本年度の報告書も非常に充実したものとなった。この報告書が、さらなる関心を引き起こすことを願っている。

２．　NTCIR-7 データを用いたドキュメント単位と文単位に基づく翻訳自動評価規準のメタ評価

北海学園大学　越前谷　博
沖電気工業株式会社　下畑　さより

2.1.1　はじめに

我々は従来より、IMPACT[1]を始めとした様々な翻訳自動評価規準のメタ評価を文単位に着目して行ってきた[2][3]。これは人手評価が文単位で行われていることが大きな理由である。このような観点より、翻訳自動評価規準により得られる文単位のスコアと文単位の人手評価との間の相関を求めることで、翻訳自動評価規準に対するメタ評価を行ってきた。文単位のメタ評価は翻訳自動評価システムの開発者から見た場合、翻訳自動評価規準の性能を検証するうえで非常に有益な情報をもたらす。

しかし、機械翻訳システムの開発者の立場としては、文単位のスコアが提示されても開発した機械翻訳システムが他の機械翻訳システムに比べ、どれだけ良い性能を有しているのかを知ることはできない。そのため、本来の自動評価システムの利用目的の観点からは文単位ではなくドキュメント単位でのスコアを得ることの方が重要となる。ドキュメント単位でのスコアでは、個々の機械翻訳システムに対して一つのスコアが出力されるため、自ら開発した機械翻訳システムと他の機械翻訳システムとの間の優劣を知るうえで利用価値が高い。

したがって、機械翻訳システムの開発者の立場としてはドキュメント単位での評価精度が重要となり、翻訳自動評価システムの開発者の立場としては文単位の評価精度が重要と考えられる。そこで、本報告では、ドキュメント単位と文単位の両方を考慮した翻訳自動評価規準のメタ評価を行った。その際には、これまでの文単位でのメタ評価において、高い精度を示した IMPACT、ROUGE-L[4]、そして、WER[5]を用いた。更に、本報告では、IMPACT のドキュメント単位と文単位のスコアの算出方法についてもその詳細を述べる。3 つの翻訳自動評価規準を用いたメタ評価の結果、ドキュメント単位の相関係数と文単位の相関係数の F 値において IMPACT が最も高い値を示した。

2.1.2　翻訳自動評価規準 IMPACT

2.1.2.1　文単位のスコア

翻訳自動評価規準 IMPACT では、BLEU[6]や NIST[7]などの多くの翻訳自動評価規準と同様、機械翻訳システムが出力する評価対象文と人手による参照訳を比較することによりスコアを求める。IMPACT は、評価対象文と参照訳間の共通部分列を一意に決定する。そして、その処理を再帰的に行うことで、語順を考慮したスコアを導き出す。共通部分列の一意の決定は以下の式（1）と（2）を用いて行う。

$$pos = (1.0 - |\frac{pos(w_m)}{m} - \frac{pos(w_n)}{n}|) \leq 1.0 \quad (1)$$

$$RS = \sum_{c \in LCS} (length(c)^{\beta} \times pos) \quad (2)$$

式（1）の m、n は参照訳と評価対象文の構成単語数をそれぞれ表している。$pos(w_m)$、$pos(w_n)$ は共通部分の先頭単語が参照訳及び評価対象文の先頭単語からどの位置に存在しているかを表している。したがって、pos は共通部分の相対位置のズレを表している。

式（2）の c は任意の共通部分、LCS は参照訳及び評価対象文中の全共通部分を表している。全共通部分の決定は、最長共通部分列（Longest Common Subsequence: LCS）に基づき決定する。$length(c)$ は共通部分の構成単語数を表している。β は共通部分の構成単語数、すなわち、共通部分の長さに対する重みづけパラメータであり、1.0 以上を用いる。pos は式（1）より得られる値である。式（1）、（2）を用いて参照訳と評価対象文間の共通部分列が一意に決定された具体例を図 1 に示す。

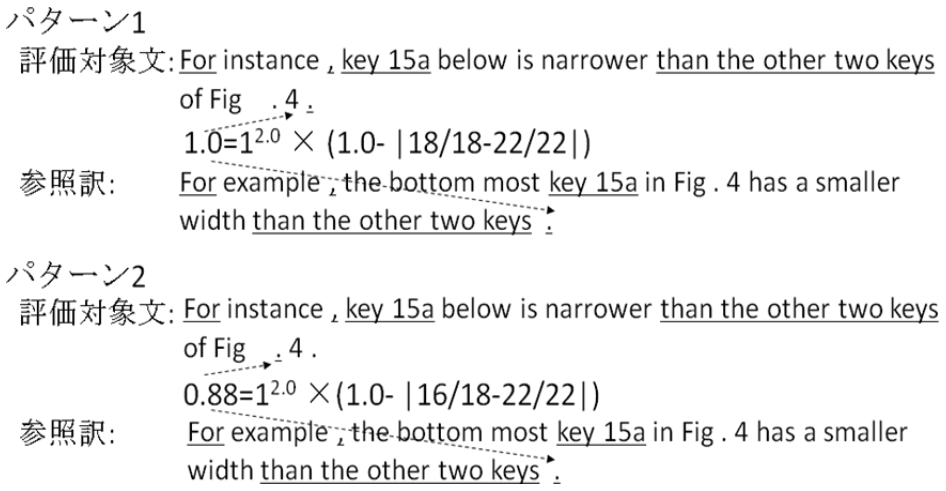


図 1　最長共通部分列の決定の具体例

図 1 では評価対象文と参照訳間の共通部分として “For”、“,”、“key 15a”、“than the other two keys”、“.” が選択された。これらの共通部分の種類はパターン 1 とパターン 2 で同じである。しかし、共通部分 “.” は文中での意味がパターン 1 と 2 で異なっている。パターン 1 においては “.” は評価対象文、参照訳共に文末のピリオドとなっている。それに対し、パターン 2 では、評価対象文においては “Figure” の省略形の “Fig.” のピリオドとなっている。したがって、図 1 の例では、パターン 1 に示された共通部分列が一意に選択されることが望ましい。そこで、IMPACT では式（1）と（2）を用いてパターン 1 を選択する。その際、他の共通部分に関しては多義性はなく、パターン 1 と 2 で差はないため、共通部分 “.” のみにより決定される。ここで、式（2）のパラメータ β の値は 2.0 である。計算の結果、パターン 1 の共通部分 “.” に対する RS は 1.0 となり、パターン 2 では 0.88 となる。したがって、パターン 1 と 2 の間の全ての共通部分を用い

た RS はパターン 1 の方がパターン 2 よりも大きな値となり、パターン 1 の共通部分列が一意に決定されることになる。

IMPACT では、このように決定された一意の共通部分列に基づきスコアを求める。具体的には以下の式（3）から（6）を用いる。

$$R_{seg} = \left(\frac{\sum_{i=0}^{RN} (\alpha^i \sum_{c \in LCS} length(c)^\beta)}{m^\beta} \right)^{\frac{1}{\beta}} \quad (3)$$

$$P_{seg} = \left(\frac{\sum_{i=0}^{RN} (\alpha^i \sum_{c \in LCS} length(c)^\beta)}{n^\beta} \right)^{\frac{1}{\beta}} \quad (4)$$

$$score_{seg} = \frac{(1+\gamma^2) R_{seg} P_{seg}}{R_{seg} + \gamma^2 P_{seg}} \quad (5)$$

$$\gamma = \frac{P_{seg}}{R_{seg}} \quad (6)$$

式（3）と（4）の RN は共通部分列の再帰的な決定の処理回数を表している。RN の値が大きいほど、評価対象文と参照訳間で語順の異なる共通部分が数多く存在していることを表している。α は語順の異なる共通部分列に対する重みづけパラメータであり、1.0 未満を用いる。したがって、i の値が大きくなるほど、すなわち、語順の異なる共通部分が出現するほど、それらのスコアの比重は小さくなる。その他の変数は式（1）、（2）と同様である。式（5）では式（3）と（4）より得られた R_{seg} と P_{seg} の調和平均を求めている。また、γ は式（6）より得る。式（3）から（6）を用いて得られる IMPACT における文単位のスコア計算の具体例を図 2 に示す。

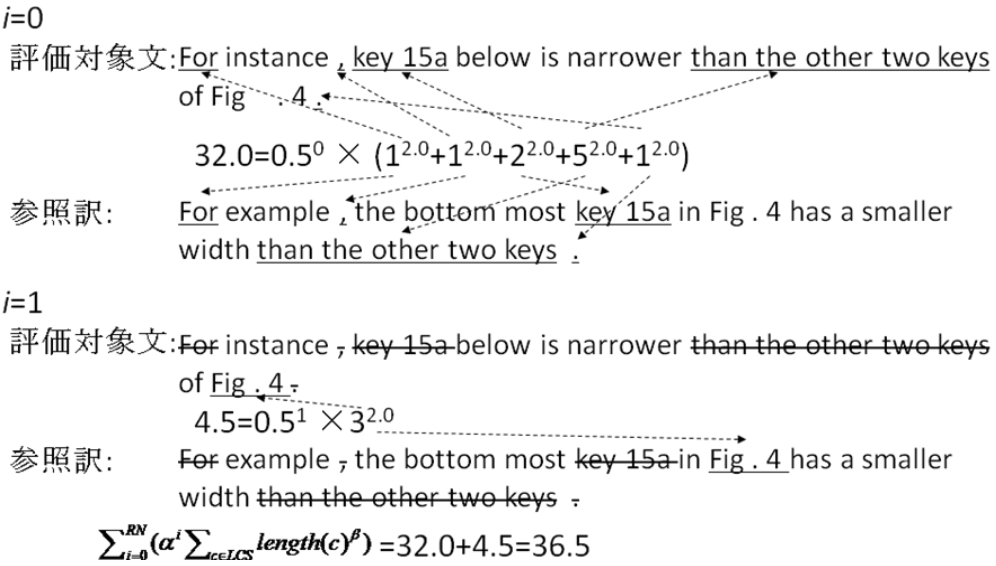


図 2　IMPACT における文単位のスコア計算の具体例

始めに、図 1 で示したパターン 1 の全ての共通部分を用いてスコアを求める。その際、式（3）と（4）の i の値は初期値の 0 である。また、パラメータ α は 0.5、β は 2.0 である。i の値が 0 の際には $\alpha^i \sum_{c \in LCS} length(c)^\beta$ の値は 32.0 となる。次いで、スコアの計算に使用された共通部分を除き、翻訳対象文と参照訳間で他に共通部分が存在している場合には、i の値をインクリメントして同様の処理を繰り返す。図 2 では共通部分として“Fig . 4”が存在するため、i の値を 1 に変化させ、処理を繰り返す。その結果、i の値が 1 の場合のスコアとして 4.5 が得られる。そして、i が 0 の場合の 32.0 に 1 の場合の 4.5 を加え、$\sum_{i=0}^{RN}(\alpha^i \sum_{c \in LCS} length(c)^\beta)$ としては 36.5 が得られる。この値に対して、参照訳と評価対象文それぞれの構成単語数、すなわち、文の長さを用いて正規化を行うことで、R_{seg} と P_{seg} を求める。その結果、$R_{seg} = \sqrt{36.6/22^{2.0}} = 0.2764$、

$P_{seg} = \sqrt{36.6/18^{2.0}} = 0.3357$ がそれぞれ得られる。この 2 つの値の調和平均を求めることで最終的な文単位のスコアとして、$score_{seg} = \frac{(1+1.2225^2)\times 0.2746\times 0.3357}{0.2764+1.2225^2\times 0.3357} = 0.2963$ が得られる。更に、複数参照訳を用いた場合のスコアの計算は以下の式（7）から（10）を用いる。

$$R_{seg-multi} = \max_{j=1}^{u}\left(\left(\frac{\left(\sum_{i=0}^{RN}(\alpha^i \sum_{c \in LCS} length(c)^\beta)\right)_j}{m_j^\beta}\right)^{\frac{1}{\beta}}\right) \qquad (7)$$

$$P_{seg-multi} = \max_{j=1}^{u}\left(\left(\frac{\left(\sum_{i=0}^{RN}(\alpha^i \sum_{c \in LCS} length(c)^\beta)\right)_j}{n_j^\beta}\right)^{\frac{1}{\beta}}\right) \qquad (8)$$

$$score_{seg-multi} = \frac{(1+\gamma^2)R_{seg-multi}P_{seg-multi}}{R_{seg-multi}+\gamma^2 P_{seg-multi}} \qquad (9)$$

$$\gamma = \frac{P_{seg-multi}}{R_{seg-multi}} \qquad (10)$$

複数参照訳を用いる場合は、個々の参照訳と評価対象文との間で得られた R_{seg} と P_{seg} の最大値を用いてスコアを得る。

2.1.2.2 ドキュメント単位のスコア

IMPACT におけるドキュメント単位でのスコアの計算方法について述べる。共通部分列の一意の決定処理については文単位の場合と同じである。ただし、最終的なスコアの計算式が文単位の場合と異なる。ドキュメント単位のスコアの計算は以下の式（11）から（14）を用いて行う。

$$R_{doc} = \left(\frac{\sum_{i=0}^{RN}\alpha^i(\sum_{c \in LCS} length(c))^\beta}{\left(\sum_{k=1}^{NS} m_k\right)^\beta}\right)^{\frac{1}{\beta}} \qquad (11)$$

$$P_{doc} = \left(\frac{\sum_{i=0}^{RN} \alpha^i (\sum_{c \in LCS} length(c))^\beta}{\left(\sum_{k=1}^{NS} n_k\right)^\beta} \right)^{\frac{1}{\beta}} \qquad (12)$$

$$score_{doc} = \frac{(1+\gamma^2) R_{doc} P_{doc}}{R_{doc} + \gamma^2 P_{doc}} \qquad (13)$$

$$\gamma = \frac{P_{doc}}{R_{doc}} \qquad (14)$$

式（11）と（12）の NS はドキュメントを構成するセグメントの数である。他の変数については式（3)、(4）と同様である。式（11）から式（12）を用いた場合のスコアの計算の具体例を図3と4に示す。図3は式（11）と（12）の i の値が0、図4は i の値が1の場合である。

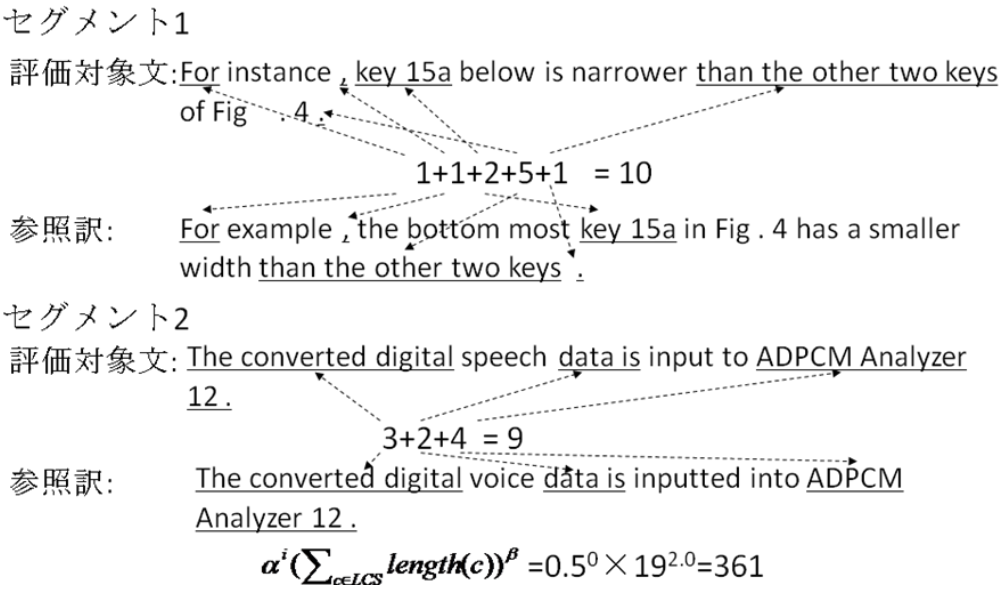


図3　IMPACTにおけるドキュメント単位のスコア計算の具体例（i=0）

式（11）と（12）の i の値が0の場合、決定された共通部分の構成単語数の総和に基づいて $\alpha^i(\sum_{c \in LCS} length(c))^\beta$ は図3より361となる。更に、図3で用いた共通部分を除き、改めて共通部分列を一意に決定する。その際には i の値は1となる。その結果、図4に示すように、$\alpha^i(\sum_{c \in LCS} length(c))^\beta$ は4.5となる。

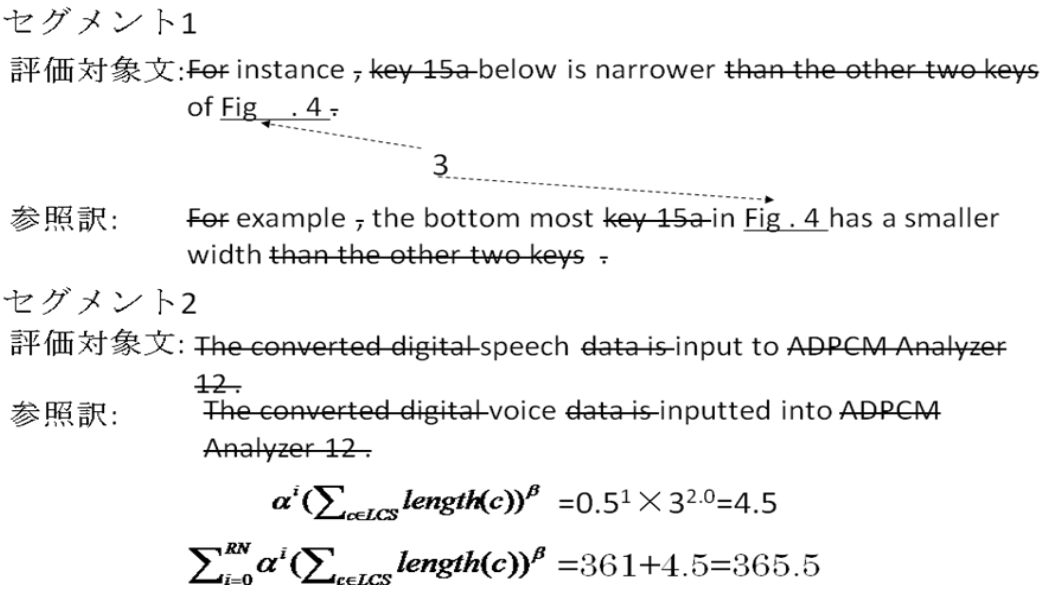


図 4　IMPACT におけるドキュメント単位のスコア計算の具体例（i=1）

図 3 と図 4 より得られたそれぞれのスコア 361 と 4.5 の和より、$\sum_{i=0}^{RN} \alpha^i(\sum_{c\in LCS} length(c))^\beta$ として 365.5 が得られる。この値に対して、参照訳と評価対象文それぞれの構成単語数の総和、すなわち、文の長さの総和を用いて正規化を行うことで、R_{doc} と P_{doc} を求める。その結果、

$R_{doc} = \sqrt{365.5/(22+13)^{2.0}} = 0.5463$、$P_{doc} = \sqrt{365.5/(18+12)^{2.0}} = 0.6373$がそれぞれ得られる。

この 2 つの値の調和平均を求めることで式（13）の最終的なドキュメント単位のスコアとして、$score_{doc} = \frac{(1+1.1666^2)\times 0.5463\times 0.6373}{0.5463+1.1666^2\times 0.6373} = 0.5815$ が得られる。更に、複数参照訳を用いた場合のスコアの計算は以下の式（15）から（18）を用いる。

$$R_{doc-multi} = \max_{j=1}^{u}\left(\left(\frac{\left(\sum_{i=0}^{RN}\alpha^i(\sum_{c\in LCS} length(c))^\beta\right)_j}{\left(\left(\sum_{k=1}^{NS} m_k\right)^\beta\right)_j}\right)^{\frac{1}{\beta}}\right) \quad (15)$$

$$P_{doc-multi} = \max_{j=1}^{u}\left(\left(\frac{\left(\sum_{i=0}^{RN}\alpha^i(\sum_{c\in LCS} length(c))^\beta\right)_j}{\left(\left(\sum_{k=1}^{NS} n_k\right)^\beta\right)_j}\right)^{\frac{1}{\beta}}\right) \quad (16)$$

$$score_{doc-multi} = \frac{(1+\gamma^2)R_{doc-multi}P_{doc-multi}}{R_{doc-multi}+\gamma^2 P_{doc-multi}} \quad (17)$$

$$\gamma = \frac{P_{doc-multi}}{R_{doc-multi}} \quad (18)$$

2.1.3 性能評価実験

2.1.3.1 実験データ

本報告では、ドキュメント単位と文単位の両方に着目し、翻訳自動評価規準のメタ評価を行った。その際の評価対象文には NTCIR-7 で使用された 15 の機械翻訳システムがそれぞれ日本語特許文 100 文を翻訳した結果である英文 100 文を用いた。また、参照訳には NTCIR-7 データに含まれている 1 つの参照訳及びそれに 3 つの参照訳を加えた 4 つの参照訳を準備した。更に、人手評価は 3 名の評価者が 5 段階評価を行った結果を用いた。

2.1.3.2 実験方法

文単位のメタ評価においては、IMPACT、ROUGE-L、WER の 3 つの翻訳自動評価規準それぞれが 15 の機械翻訳システムより出力された全評価対象文 1,500 文に対して、参照訳を用いて文単位のスコアを計算する。また、人手評価においては評価対象文ごとに 3 名の評価者による評価値の平均を求める。そして、翻訳自動評価規準により得られた 1,500 のスコアと人手評価の 1,500 の値との間の相関を求める。具体的には、ピアソンの相関係数とスピアマンの順位相関係数を求める。また、ドキュメント単位のメタ評価においては、3 つの翻訳自動評価規準それぞれが翻訳対象文と参照訳を用いて機械翻訳システムごとにスコアを計算する。また、人手評価においては機械翻訳システムごとに 3 名の評価者による評価値の平均を求め、それらを人手評価の値として用いる。そして、翻訳自動評価規準により得られた 15 のスコアと人手評価の 15 の値を用いてピアソンの相関係数とスピアマンの順位相関係数を求める。更に、文単位の相関係数とドキュメント単位の相関係数を用いて F 値を求める。

2.1.3.3 実験結果

表 1 から表 4 に 1 つの参照訳を用いた場合の結果を示す。一般的には、常に多くの参照訳を得ることが困難であるため 1 つの参照訳を用いた際のメタ評価の結果は重要と考えられる。その際には NTCIR-7 データ[8]に含まれている参照訳を用いた。表 1 に Adequacy におけるピアソンの相関係数、表 2 に Fluency におけるピアソンの相関係数、表 3 に Adequacy におけるスピアマンの順位相関係数、表 4 に Fluency におけるスピアマンの順位相関係数をそれぞれ示す。また、表 5 から表 8 には 4 つの参照訳を用いた場合の結果を示す。表中の“No.1”から“No.15”は 15 の機械翻訳システムそれぞれの文単位の相関係数である。“Seg.”は 2.1.3.2 節で述べた全翻訳対象文を用いた場合の文単位の相関係数である。“Doc.”は、ドキュメント単位の相関係数である。

表 1 から表 8 より、表 1 と表 5、すなわち、Adequacy におけるピアソンの相関係数以外では全て IMPACT の F 値が最も高かった。特に、文単位の相関係数が表 1 から表 8 の全てで最も高く、そのことが IMPACT の F 値が高くなった要因となった。表 4 の文単位の相関係数においては、IMPACT と ROUGE-L の差は 0.5%未満の有意水準で有意であった。しかし、ドキュメント単位においては、IMPACT は表 1 から表 8 の全てで最も高い相関係数を示すことはなく今後の課題である。

表 1　1 つの参照訳を用いた Adequacy におけるピアソンの相関係数

	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
IMPACT	0.7221	0.3068	0.3802	0.4380	0.4945	0.4948	0.5904	0.7170	0.3144
ROUGE-L	0.7084	0.2466	0.3324	0.4242	0.4543	0.4567	0.5499	0.6984	0.3181
WER	0.6914	0.1877	0.2969	0.4021	0.4155	0.3822	0.4842	0.6861	0.4428
	No.10	No.11	No.12	No.13	No.14	No.15	Seg.	Doc.	F 値
IMPACT	0.6865	0.5321	0.6387	0.5657	0.7564	0.5440	0.6535	0.8511	0.7393
ROUGE-L	0.6646	0.4991	0.6259	0.5324	0.7412	0.5271	0.6463	0.8671	0.7406
WER	0.6252	0.4590	0.5957	0.4923	0.7443	0.4573	0.5907	0.8881	0.7095

表 2　1 つの参照訳を用いた Fluency におけるピアソンの相関係数

	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
IMPACT	0.5835	0.2228	0.3429	0.4405	0.4244	0.5557	0.5562	0.7014	0.2782
ROUGE-L	0.5719	0.1431	0.2930	0.4314	0.3749	0.5197	0.5064	0.6959	0.2602
WER	0.5217	0.1298	0.2333	0.3323	0.2999	0.4037	0.4298	0.6556	0.3054
	No.10	No.11	No.12	No.13	No.14	No.15	Seg.	Doc.	F 値
IMPACT	0.6168	0.4879	0.6122	0.3560	0.6707	0.4896	0.6484	0.9560	0.7727
ROUGE-L	0.6001	0.4484	0.6154	0.3308	0.6524	0.4754	0.6440	0.9596	0.7708
WER	0.5905	0.3924	0.5575	0.3093	0.6257	0.4012	0.5376	0.9187	0.6783

表 3　1 つの参照訳を用いた Adequacy におけるスピアマンの順位相関係数

	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
IMPACT	0.6659	0.3218	0.4209	0.4519	0.4015	0.4470	0.5200	0.6333	0.3541
ROUGE-L	0.6423	0.2726	0.3827	0.4207	0.3599	0.3954	0.4698	0.6153	0.3502
WER	0.5793	0.1645	0.3063	0.3898	0.3098	0.3162	0.3527	0.5427	0.4011
	No.10	No.11	No.12	No.13	No.14	No.15	Seg.	Doc.	F 値
IMPACT	0.6864	0.4160	0.6131	0.5184	0.7218	0.5894	0.6462	0.9321	0.7633
ROUGE-L	0.6525	0.3856	0.5890	0.4829	0.7138	0.5466	0.6337	0.9500	0.7602
WER	0.5775	0.2867	0.5180	0.4556	0.7174	0.4476	0.5466	0.9571	0.6958

表 4　1 つの参照訳を用いた Fluency におけるスピアマンの順位相関係数

	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
IMPACT	0.5912	0.1871	0.3377	0.4473	0.3806	0.5228	0.4946	0.6568	0.3074
ROUGE-L	0.5635	0.1224	0.3146	0.4161	0.3257	0.4654	0.4373	0.6573	0.2908
WER	0.4562	0.0861	0.2216	0.3140	0.2298	0.3483	0.3350	0.5955	0.2692
	No.10	No.11	No.12	No.13	No.14	No.15	Seg.	Doc.	F 値
IMPACT	0.5858	0.3854	0.6436	0.3616	0.7091	0.5751	0.6350	0.8536	0.7282
ROUGE-L	0.5607	0.3452	0.6331	0.3329	0.6956	0.5289	0.6195	0.8750	0.7254
WER	0.5396	0.2557	0.5221	0.2960	0.6724	0.4012	0.5153	0.8679	0.6466

表 5　4 つの参照訳を用いた Adequacy におけるピアソンの相関係数

	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
IMPACT	0.7962	0.5263	0.4827	0.6321	0.5911	0.6477	0.6679	0.7578	0.3279
ROUGE-L	0.7899	0.4610	0.4364	0.6410	0.5738	0.6345	0.6350	0.7386	0.3276
WER	0.7737	0.4304	0.3819	0.6013	0.5543	0.5601	0.6279	0.7301	0.4295
	No.10	No.11	No.12	No.13	No.14	No.15	Seg.	Doc.	F 値
IMPACT	0.7507	0.6214	0.7235	0.6671	0.8033	0.5767	0.7401	0.9054	0.8144
ROUGE-L	0.7330	0.6225	0.7140	0.6347	0.7823	0.5626	0.7379	0.9218	0.8196
WER	0.7194	0.5595	0.6868	0.6230	0.7930	0.5399	0.7146	0.9571	0.8183

表 6　4 つの参照訳を用いた Fluency におけるピアソンの相関係数

	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
IMPACT	0.6038	0.4003	0.4087	0.5695	0.4816	0.6159	0.5539	0.6861	0.3063
ROUGE-L	0.5917	0.3202	0.3487	0.5804	0.4477	0.5934	0.5046	0.6710	0.2845
WER	0.5596	0.2568	0.2999	0.5062	0.4212	0.5534	0.4782	0.6276	0.2847
	No.10	No.11	No.12	No.13	No.14	No.15	Seg.	Doc.	F 値
IMPACT	0.6612	0.5963	0.6740	0.4522	0.7074	0.5105	0.7104	0.9710	0.8205
ROUGE-L	0.6400	0.5897	0.6749	0.4227	0.6836	0.4967	0.7076	0.9737	0.8196
WER	0.6407	0.5377	0.6598	0.4520	0.6968	0.4745	0.6434	0.9371	0.7630

表 7　4 つの参照訳を用いた Adequacy におけるスピアマンの順位相関係数

	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
IMPACT	0.7681	0.4643	0.5395	0.6452	0.5176	0.6324	0.6095	0.6682	0.3463
ROUGE-L	0.7593	0.4231	0.4990	0.6538	0.5075	0.6191	0.5630	0.6565	0.3374
WER	0.6683	0.3957	0.3659	0.6055	0.4592	0.5459	0.5076	0.5886	0.3879
	No.10	No.11	No.12	No.13	No.14	No.15	Seg.	Doc.	F 値
IMPACT	0.7238	0.5344	0.7187	0.6112	0.7794	0.6093	0.7415	0.9821	0.8450
ROUGE-L	0.7133	0.5445	0.7098	0.5797	0.7563	0.5650	0.7357	0.9857	0.8425
WER	0.6854	0.4124	0.6630	0.5678	0.7523	0.5723	0.6819	0.9929	0.8085

表 8　4 つの参照訳を用いた Fluency におけるスピアマンの順位相関係数

	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
IMPACT	0.5953	0.3398	0.4148	0.5502	0.4367	0.6121	0.5020	0.6380	0.2844
ROUGE-L	0.5829	0.2714	0.3606	0.5595	0.4027	0.5915	0.4421	0.6203	0.2684
WER	0.5000	0.2036	0.2551	0.4894	0.3333	0.5591	0.3908	0.5625	0.2956
	No.10	No.11	No.12	No.13	No.14	No.15	Seg.	Doc.	F 値
IMPACT	0.6410	0.5171	0.7034	0.4480	0.7186	0.5755	0.6933	0.9071	0.7859
ROUGE-L	0.6196	0.5230	0.6939	0.4118	0.6836	0.5266	0.6819	0.9036	0.7773
WER	0.6119	0.4127	0.6733	0.4323	0.7106	0.5069	0.6322	0.9286	0.7522

2.1.4 まとめと今後の予定

本報告では文単位とドキュメント単位の両方の観点に基づき翻訳自動評価規準のメタ評価を行った。文単位での評価精度は翻訳自動評価システムの開発者の観点から重要であり、ドキュメント単位での評価精度は機械翻訳システムの開発者の観点より重要と考えられる。IMPACT、ROUGE-L、WER の 3 つの翻訳自動評価規準を用いたメタ評価の結果、IMPACT が最も高い評価精度を示した。特に、文単位のメタ評価において IMPACT は高い相関を示した。しかし、ドキュメント単位では、IMPACT の結果は十分とはいえない。

今後は、IMPACT におけるドキュメント単位での評価精度の向上を図る。また、他の言語や分野のデータを用いた更なるメタ評価を行っていく予定である。

謝辞

この研究は国立情報学研究所との共同研究に関連して行われた。

参考文献

[1] Echizen-ya, Hiroshi. and Araki, Kenji. Automatic Evaluation of Machine Translation based on Recursive Acquisition of an Intuitive Common Parts Continuum, Proceedings of the Eleventh Machine Translation Summit, pp.151-158, 2007.

[2] 越前谷博、江原暉将、下畑さより、藤井敦、内山将夫、山本幹雄、宇津呂武仁、神門典子. NTCIR-7 データを用いた機械翻訳自動評価基準のメタ評価, 平成 20 年度 AAMT/Japio 特許翻訳研究会報告書, pp.2-13, 2009.

[3] Echizen-ya, Hiroshi., Ehara, Terumasa., Shimohata, Sayori., Fujii, Atsushi., Utiyama, Masao., Yamamoto, Mikio., Utsuro, Takehito. and Kando, Noriko. Meta-Evaluation of Automatic Evaluation Methods for Machine Translation using Patent Translation Data in NTCIR-7, Proceedings of the 3rd Workshop on Patent Translation, pp.9-16, 2009.

[4] Lin, Chin-Yew. and Och, Franz Josef. Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics, Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics, pp.606-613, 2004.

[5] Leusch, Gregor., Ueffing, Nicola. and Ney, Hermann. A Novel String-to-String Distance Measure With Applications to Machine Translation Evaluation, Proceedings of the 9th Machine Translation Summit, pp.240-247, 2003.

[6] Papineni, Kishore., Roukos, Salim., Ward, Todd. and Zhu, Wei-Jing. BLEU: a Method for Automatic Evaluation of Machine Translation, Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, pp.311-318, 2002.

[7] NIST. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics, 2002, http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf

[8] Fujii, Atsushi., Utiyama, Masao., Yamamoto, Mikio. and Utsuro, Takehito. Overview of the Patent Translation Task at the NTCIR-7 Workshop. Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access, pp.389-400, 2008.

３．１　対訳特許文を用いた同義対訳専門用語収集手法およびその評価

筑波大学大学院システム情報工学研究科
梁　冰，宇津呂　武仁，山本　幹雄

3.1.1 はじめに

特許文書の翻訳は,他国への特許申請や特許文書の言語横断検索などといったサービスにおいて不可欠である．特許文書翻訳の過程において，専門用語の対訳辞書は重要な情報源であり，これまでに，対訳特許文書を情報源として，専門用語対訳対を自動獲得する手法の研究が行われてきた．文献［3］では，NTCIR-7 の特許翻訳タスク［1］で配布された日英 180 万件の対訳特許文を用いて，対訳特許文からの専門用語対訳対獲得を行った．この研究では，句に基づく統計的機械翻訳モデル［2］を用いることにより日英対訳文から学習されたフレーズテーブル，要素合成法［5］，Support Vector Machines (SVMs)［7］による機械学習を用いることによって,専門用語対訳対獲得における適合率を改善している.この手法の問題点の一つとして,ある日本語専門用語に対する英訳語を推定する際に,その日本語専門用語が出現する一つの対訳文に出現する英訳語のみを推定対象とする点が挙げられる．そのため，この手法では，ある日本語専門用語が出現する複数の対訳文を入力として，同義・異義となる専門用語対訳対の集合を同定することができない．そこで本論文では,ある日本語専門用語が出現する複数の対訳文を入力として英訳語の推定を行うことにより，同義となる専門用語対訳対を同定することを目的とする．本論文において提案する手順においては，まず，ある日本語専門用語が出現する複数の対訳文を入力として，同義となる専門用語対訳対の候補を生成する．生成した候補集合の中から同義判定するための中心的対訳対を選び，中心的対訳対のうちの日本語専門用語に対して，専門用語対訳対同義候補集合を再生成する．再生成した候補集合に対して SVM を適用することにより，同義集合・異義集合を同定する．評価実験においては，同義判定において 97.4% の適合率を達成した．

3.1.2 日英対訳特許文

本論文では,NTCIR-7 の特許翻訳タスク［1］で配布された約 180 万対の日英文対応データを，フレーズテーブルの訓練用データとして使用した．この文対応データは，1993-2000 年発行の日本公開特許広報全文と米国特許全文を対象として，文献［6］によって日英間で文対応を付けたものである．

3.1.3 句に基づく統計的機械翻訳モデルのフレーズテーブル

本論文では，文献［3］の場合と同様に，専門用語の訳語推定において，日英対訳特許文から学習したフレーズテーブルを用いる．フレーズテーブルにおいては，2 節で述べた文対応データに対して，句に基づく統計的機械翻訳モデルのツールキットである Moses［2］を適用することにより，日英の句の組，及び，日英の句が対応する確率を推定し記述する．Moses によってフレーズテーブルを作成する過程を以下に示す．

1. 文対応データに対する前処理として，単語の数値化，単語のクラスタリング，共起単語表の作成などを行う．

2. 文対応データから最尤な単語対応を英日，日英の両方向において得る．
3. 英日，日英両方向の単語対応から，ヒューリスティックスを用いて対称な単語対応を得る．
4. 対称な単語対応を用いて，可能な全ての日英の句の組を作成し，各組に対して，「文単位の句対応制約」の条件に対する違反の有無をチェックする(違反しない句の組を有効な対応とみなす)．
5. 文対応データにおける日英の句の対応の数を集計し，各句の対応に翻訳確率を付与する．

手順 4 について，以下に「文単位の句対応制約」の条件を示す．

日本語文の形態素列中の形態素を文頭から順に $V_1, V_2, \cdots, V_n$ 英文の単語列中の単語を文頭から順に $W_1, W_2, \cdots, W_m$ として，日本語句を $P_J (= V_p \cdots V_{p'})$ とし，英語句を $P_E (= W_q \cdots W_{q'})$ とする．ここで，日英句の組 $< P_J, P_E >$ が含まれるある一つの対訳文対 $\langle T_J, T_E \rangle$ 中において得られているあらゆる単語対応 $\langle V_i, W_j \rangle$ について，が成り立つ場合に，P_J と P_E は対訳文対 $\langle T_J, T_E \rangle$ において「文単位の句対応制約」に違反しない，と定義する．

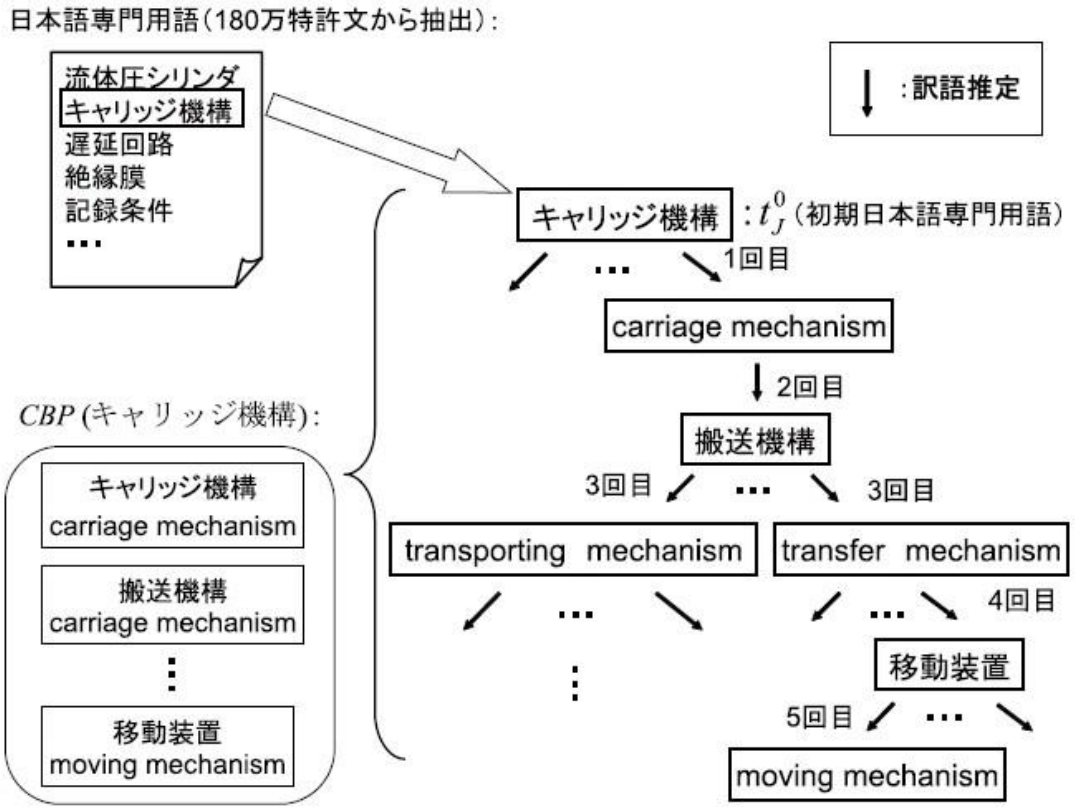


図 1: 専門用語対訳対同義候補集合の作成

表 1: 作成された専門用語対訳対同義候補集合中の対訳対数

	総要素数	134 個の集合の間の平均対数
同義候補集合 $\bigcup_{s_J} CBP(s_J)$	22,473	167.7
人手で同定した同義集合 $\bigcup_{s_{JE}} SBP(s_{JE})$	1,680	12.5

3.1.4 フレーズテーブルを用いた専門用語対訳対の同義集合の生成

3.1.4.1 専門用語対訳対同義候補集合の作成

図 1 に，専門用語対訳対同義候補集合作成の流れを示す.

1. 180 万文の特許文から無作為に抽出した初期日本語専門用語 t_J^0 に対し，全対訳特許文 180 万件から学習されたフレーズテーブル[1]を用いて訳語推定を行い，英語訳語を得る.
2. 1 で得られた英語専門用語に対して訳語推定を行い，日本語訳語を得る.
3. 1，2 の手順を繰り返し，k 回訳語推定を行うことにより得られた対訳専門用語を集めた集合を $CBP(t_J^0)$ とする（本論文では，k = 6 とした）.

本論文では，以上の手順に従って，4,000 個の初期日本語名詞句を用いて，専門用語対訳対の候補集合 $CBP(t_J^0)$ を作成した．なお，本論文では，専門用語対訳対同義候補集合 $CBP(t_J^0)$ に対して，要素数の下限を設定した（具体的には，$|CBP(t_J^0)| \geq 10$）.

3.1.4.2 中心的対訳対を用いた参照用同義集合の作成

次に，前節で作成した同義候補集合 $CBP(t_J^0)$ 中の専門用語対訳対の中から，「一般語の対訳対」を除いて，180 万対訳文中の頻度が最大となる対訳対を選定し，**中心的対訳対** $s_{JE} = \langle s_J, s_E \rangle$ とする[2]．ここで，本論文では，対訳対が以下の条件を全て満たす場合に，その対訳対は「一般語の対訳対」であるというヒューリスティクスを用いた.

1. 180 万対訳文における頻度が500 以上．2. 日本語用語が以下のいずれかを満たす.
   (a) 漢字または平仮名を含む場合は，二文字以下.
   (b) カタカナ語の場合は，複合語でない.
3. 英語用語が一単語.

以上の手順に従って，合計150 個の中心的対訳対を選定した．次に，中心的対訳対 s_{JE} の

[1]ただし，日英方向の訳語推定を行う場合は，日英方向のフレーズテーブルの順位が一位となる英訳語を用い，英日方向の訳語推定を行う場合は，英日方向のフレーズテーブルの順位が一位となる日本語訳語を用いた．また，対訳特許文180 万件中の頻度が6 以上800 以下となる対訳対に限定した．なお，フレーズテーブルを用いた日英方向の訳語推定の精度は，91.9%である[3].

うちの日本語専門用語 s_J を用いて，前節の手順によって専門用語対訳対同義候補集合 $CBP(s_J)$ を作成する．ただし，以下では，要素数の下限（具体的には，$|CBP(s_J)| \geq 10$ を満たすもののみを対象とする．

以上の手順の結果，134 個の専門用語対訳対同義候補集合が作成された．表1 に示すように，専門用語対訳対の総数は，22,473 個となった，なお，この過程において，訳語対応として正しくない対訳対は人手で除外した．

最後に，人手によって，同義候補集合 $CBP(s_J)$ を，中心的対訳対 sJE と同義となる対訳対の集合 $SBP(s_{JE})$，および，その他の対訳対の集合 $NSBP(s_{JE})$ に分割する．この結果，表1 に示すように，中心的対訳対と同義となる対訳対の総数は1,680個となった．

表 2：専門用語対訳対の同義・異義集合同定のための素性

分類	素性名	定義 （ただし，$X \in \{J,E\}$，$(Y,Z) \in \{(J,E),(E,J)\}$）
対訳対 $\langle t_J, t_E \rangle$ の特性を規定する	f_1: 出現頻度	対訳特許文における $\langle t_J, t_E \rangle$ の出現頻度の自然対数．
	f_2: 英語訳語の順位	条件付き確率 $P(t_E \mid t_J)$ の降順に t_E を順位付けしたときの t_E の順位．
	f_3: 日本語訳語の順位	条件付き確率 $P(t_J \mid t_E)$ の降順に t_J を順位付けしたときの t_J の順位．
	f_4: 日本語文字数	t_J の文字数．
	f_5: 英語単語数	t_E の単語数．
	f_6: 訳語推定における繰り返しの回数	s_J から訳語推定を開始し，訳語として t_Y を生成した直後に t_Y から t_Z を訳語推定した場合の，s_J から t_Z までの繰り返し訳語生成回数．
対訳対 $\langle t_J, t_E \rangle$ と中心的対訳対 $\langle s_J, s_E \rangle$ の間の関係を規定する	f_7: 日本語用語が同一	$t_J = s_J$ ならば，1 となる．
	f_8: 英語用語が同一	$t_E = s_E$ ならば，1 となる．
	f_9: 編集距離類似度	$f_9(t_X, s_X) = 1 - \frac{ED(t_X, s_X)}{\max(\lvert t_X \rvert, \lvert s_X \rvert)}$ ：ED は t_X と s_X の間の編集距離，$\lvert t \rvert$ は t に含まれる文字数を表す．
	f_{10}: バイグラム類似度	$f_{10}(t_X, s_X) = \frac{\lvert bigram(t_X) \cap bigram(s_X) \rvert}{\max(\lvert t_X \rvert, \lvert s_X \rvert)+1}$ ：$bigram(t)$ は，t に含まれる文字単位のバイグラムの集合．
	f_{11}: 同一の形態素・単語数の割合	$f_{11}(t_X, s_X) = \frac{\lvert const(t_X) \cap const(s_X) \rvert}{\max(\lvert const(t_X) \rvert, \lvert const(s_X) \rvert)}$ ：$const(t)$ は t に含まれる形態素または単語の集合．
	f_{12}: 日本語用語の文字列の包含関係もしくは異表記	t_J と s_J は，以下のいずれかの関係を満たす．(i) 構成要素の差分は接尾辞のみ，(ii) 構成文字列の差分は，長音「ー」のみ，(iii) 構成文字列の差分は，送り仮名の違いのみ．
	f_{13}: 英語語幹が同一	t_E と s_E の構成単語数が同一，かつ，対応する位置の単語の語幹が同一となる．
	f_{14}: 英語用語のハイフン・スペース	t_E と s_E は，ハイフンまたはスペースの有無のみが異なる．
	f_{15}: 非共有箇所に対し要素合成法の同一訳が存在	t_X, s_X で文字列が一致しない箇所 x_i, x_j に対して，要素合成法による訳語推定を行った場合に，同一訳が存在する．
	f_{16}: 要素合成法の共通訳が存在	要素合成法により，t_Y を訳語推定し s_Z が得られる．または s_Z を訳語推定し t_Y が得られる．
	f_{17}: フレーズテーブルの共通訳が存在	フレーズテーブルにより，t_Y を訳語推定し s_Z が得られる．または s_Z を訳語推定し t_Y が得られる．

3.1.5 同義・異義判定のための素性

同義専門用語対訳対の同定に用いた素性を表2 に示す．素性は大きく，対訳対 $\langle t_J, t_E \rangle$ の特性を規定するもの，および，対訳対 $\langle t_J, t_E \rangle$ と中心的対訳対 $\langle s_J, s_E \rangle$ の間の関係を規定するものの2 種類に分けられる．

[2]我々が行った先行研究[4] においては，専門用語対訳対同義候補集合中の全ての同義組および異義組を同定するタスクを設定した．一方，本節では，専門用語対訳対同義候補集合中において中心的対訳対を選定し，中心的対訳対との間でのみ同義・異義を識別するという，より簡単化したタスクを設定する点が異なる．本節においては，要素技術の性能の限界を解明することを主目的として，問題の本質を特定するためのタスク設定を採用した．

表 3: 同義判定の性能評価(%)

手法		適合率	再現率	F 値
ベースライン		67.0	54.3	60.8
SVM	適合率最大	**97.4**	31.0	44.9
	F 値最大	73.3	62.9	**65.7**

表 4: 同義判定における SVM による改善例

ベースライン:	t_J と s_J が同一，または， t_E と s_E が同一
SVM:	適合率が最大となる 下限を用いたモデル

(a) SVM のみで同義と判定し正解

$\langle t_J, t_E \rangle - \langle s_J, s_E \rangle$	人手による同義・異義判定	ベースラインによる判定	SVMによる判定
〈保持回路，holding circuit〉−〈ホールド回路，hold circuit〉	同義	異義	同義

(b) ベースラインのみで同義と判定し不正解

$\langle t_J, t_E \rangle - \langle s_J, s_E \rangle$	人手による同義・異義判定	ベースラインによる判定	SVMによる判定
〈搬送ユニット，transfer unit〉−〈転写器，transfer unit〉	異義	同義	異義

3.1.6 機械学習を用いた同義・異義判定

3.1.6.1 適用手順

前節で示した素性を用いて，SVM による同義・異義判定の評価を行った．4.2 節において作成した専門用語対訳対同義候補集合$CBP(s_J)$を全参照用事例として，8 割を用いて SVM の訓練を行い，残りのうちの1 割を用いて2 種類のパラメータの調整を行い，最後の 1 割を評価用事例とした．以上の手順を10 通り繰り返し，その平均値を算出し同義判定の性能評価を行った．2 種類のパラメータの調整においては，同義判定の適合率を最大化する場合，および，同義判定のF 値を最大化する場合の2 通りの調整を行った．なお，本 3.1節で調整の対象としたパラメータは，SVM のソフトマージンを制約するパラメータ，および，分離平面から評価用事例までの距離の下限である．

3.1.6.2 評価結果

表3 に，同義判定における性能の評価結果を示す．ベースラインとしては，「t_J と s_J が同一，または，t_E と s_E が同一」という条件を用いた．同義判定の適合率を最大化する

調整を行った場合は，97.4%の適合率を達成した．一方，同義判定のF 値が最大化する調整を行った場合は，ベースラインのF 値を上回るF 値を達成した．表4 に，SVM による同義判定の改善例を示す．「(a)SVM のみで同義と判定し正解」の例においては，「英語語幹が同一」，「非共有箇所に対し要素合成法の同一訳が存在」，「フレーズテーブルの共通訳が存在」等の素性の効果によって,SVM のみが同義と判定できたと考えられる.一方，「(b) ベースラインのみで同義と判定し不正解」の例においては，「日本語編集距離類似度」や「要素合成法の共通訳が存在」等の素性の効果によって，SVM のみが異義と判定できたと考えられる.

3.1.7 おわりに

本節では,対訳特許文を用いて,同義対訳専門用語の同定と収集を行う手法を提案した.特に，同義・異義判定のための素性を用いて，SVM によって同義・異義判定を行う手法を提案した．評価実験においては，97%以上の適合率を達成した．今後は，SVM によって高適合率で同義と判定した専門用語対訳対を自動的に中心的対訳対として選定し,専門用語対訳対同義候補集合の作成と SVM の適用を再帰的に行う枠組みを開発し，再現率を改善する方式について研究を進める.

参考文献

1] A. Fujii, M. Utiyama, M. Yamamoto, and T. Utsuro.Overview of the Patent Translation Task at the NTCIR-7Workshop. In *Proc. 7th NTCIR Workshop Meeting*, pp.389-400, 2008.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico,N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens,C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses:Open source toolkit for statistical machine translation. In*Proc. 45th ACL, Companion Volume*, pp. 177-180, 2007.

[3] 森下洋平，梁冰，宇津呂武仁，山本幹雄．フレーズテーブルおよび既存対訳辞書を用いた専門用語の訳語推定．電子情報通信学会論文誌，Vol. J93-D，No. 11，pp. 2525-2537，2010.

[4] 森下洋平，宇津呂武仁，山本幹雄．対訳特許文からの対訳専門用語獲得における同義専門用語集合の分析と同定．言語処理学会第16 回年次大会論文集，pp. 206-209，2010.

[5] 外池昌嗣，宇津呂武仁，佐藤理史．ウェブから収集した専門分野コーパスと要素合成法を用いた専門用語訳語推定．自然言語処理，Vol. 14，No. 2，pp. 33-68，2007.

[6] M. Utiyama and H. Isahara. A Japanese-English patentparallel corpus. In *Proc. MT Summit XI*, pp. 475-482,2007.

[7] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

３．２　素性間の類似度による同義語抽出への影響に関する調査

東京大学　範　暁蓉
愛媛大学　二宮　崇

3.2.1　はじめに

同じ意味を共有する異なる表記の語は同義語と呼ばれ、数多くの同義語が存在することが知られている。自然言語処理では同義語の自動獲得が非常に重要な役割を担っている。たとえば検索システムにおいて同義語をあらかじめ特定していれば、検索クエリに対する同義語で検索した結果もユーザに返すことができる。コーパスから同義語を獲得する従来技術は、二つの単語間の分布類似度（たとえば、Dice 係数）を必要とする。すなわち、単語の前後の文脈や係り受け関係を素性として単語間の類似度を計算して、類似度がある閾値以上の単語対は同義語とする。しかし、従来の研究では表記が同じ素性だけが共通素性として用いられ、表記が異なるが意味が同じと考えられる素性は共通素性として用いられていなかった。

本研究では、二つの素性間の類似度を用いて、表記が異なる素性を共通素性とみなす手法を提案する。この手法を中国語における同義語抽出に適用し、実験を行った結果を報告する。

本稿の構成は以下のようになっている。3.2.2 節では、従来の分布類似度に基づいた同義語獲得技術を説明する。3.2.3 節は、素性間の類似度による同義語抽出の手法を説明し、3.2.4 節では、提案手法でコーパスからの同義語抽出の実験・結果について報告する。3.2.5 節で本稿の主旨をまとめ、今後の課題について述べる

3.2.2　分布類似度に用いた同義語抽出

同義語抽出技術では、一般に「Distributional Hypothesis」という仮説に基づき、単語の類似度を計算する。本節では、分布類似度に用いた同義語抽出技術について述べる。

3.2.2.1　分布類似度

分布類似度とは「類似した文脈を持つ語は意味も類似している可能性が高い」という「分布仮説」（Harris ら, 1985）に基づいて計算される語の類似度である。対象単語の周辺に登場した単語（Curran ら, 2002）や、構文情報を用いて取り出した係り受け関係（Dekang Lin, 1998）など、様々な文脈情報は単語の特性を反映できている。有効な文脈情報を選択し素性として用いることにより同義語を高い精度で発見することが可能になる。

単語の係り先を素性とした場合を例に挙げる。たとえば、「意见（意見）」「看法（見方）」「成品（完成品）」という単語は以下のような素性を持っている。

単語候補	素性
意见	（1）阐明（2）赞成（3）提出（4）汇集（5）交流（6）发表
看法	（1）表明（2）赞同（3）提出（4）交换（5）交流（6）发表

成品	（1）制造（2）生产（3）销售（4）供应

この場合、「意见」と「看法」は、素性に共通部分が多いため、類似した文脈を持つと考えられ、類似度が高くなる。「意见」と「成品」は共通部分がなく、類似した文脈を持たないと考えられ、類似度は低くなる。このように、文脈の類似度によって、単語間の類似度を計算するのが、分布類似度という手法である。

3.2.2.2 同義語抽出

分布類似度に基づく同義語の抽出は二つのステップで計算される。まず、各単語に対して、文脈情報（係り先と係り元と対象語の前後連続単語など）を抽出して、有効な文脈情報を選択し、各単語の「素性ベクトル」を生成する。次に、生成された素性ベクトルを利用して、単語間の類似度を計算する。

前述のように、コーパス中の文脈情報を収集することで、各単語に対する素性ベクトルが求まり、二つの単語 w1,w2∈W（W はコーパス中に出現する単語の集合）に対する類似度はそれらの素性ベクトル間の類似度として計算される。類似度の計算に用いられる距離尺度はいくつか存在する。典型的な尺度として Jaccard 係数や Cosine 距離、Dice 係数などが存在する。Weeds は博士論文（Weeds, 2003）の中で、これらの類似度アルゴリズムに関する詳細な説明を与えている。本稿では Dice 係数（Curran ら, 2002）を用いて類似度の計算を説明する。

Dice†係数は次のように計算する。

$$Dice\dagger = \frac{2\sum_C \min(weight(w_1,c), weight(w_2,c))}{\sum_C weight(w_1,c) + weight(w_2,c)}$$

ただし、weight は重み関数であり、C は素性の集合であり、c は素性の一つである。

3.2.3 素性間の類似度による同義語の抽出

前述した通り、共通する素性を持つ単語は同義語の可能性が高い。しかし、人が同じ意味を表現するときには、異なる語を使うことも多い。すなわち、素性として抽出された文脈情報の間も同義語であることが多い。

単語候補	共通素性	異なる素性
意见	（3）提出（5）交流（6）发表	（1）阐明（2）赞成（4）汇集
看法	（3）提出（5）交流（6）发表	（1）表明（2）赞同（4）交换

たとえば、前述した「意见」と「看法」は三つの共通素性を持っている。単語と素性の重みをすべて 1 と仮定すれば、Dice†係数によるこの二つ単語間の類似度は 0.5 となり、「意见」と「看法」が同義語であることを判定することは難しい。しかし、異なる素性の中には同義語が存在しており、これらを共通素性とするならば、この二つの単語を同義語と判断することが可能となる。

例えば、「闡明（明らかにする）」と「表明（表明する）」は同義語であり、「赞成（賛成する）」と「赞同（賛同する）」も同義語である。もし、この二つの同義語対も共通素性と考えられれば、Dice†係数によるこの二つ単語対の類似は 0.83 になり、「意见」と「看法」が同義語であることを判定できる。

同義語辞書があれば、異なる素性の対が同義語かどうか判断できるが、汎用の同義語辞書だけでは登録語数が少ないため問題がある。そのため、同義語辞書だけでなく、素性間の類似度を使って、この問題を解決する。素性の類似度を計算するときも分布類似度を用いて計算する。

提案手法に対して計算尺度（Dice†係数）は以下のように計算する。

$$Dice\dagger* = \frac{2\sum_C \min(sim(c_1,c_2)*weight(w_1,c_1), sim(c_1,c_2)*weight(w_2,c_2))}{\sum_C sim(c_1,c_2)*weight(w_1,c_1) + sim(c_1,c_2)*weight(w_2,c_2)}$$

ただし、*sim* 関数は素性間の類似度である。

3.2.4 実験

本節では、提案手法の有効性を確認するため中国語の同義語抽出実験の手順および結果について述べる。

3.2.4.1 コーパス

実験用コーパスは、4 年分の Chinese Giga Word Third Edition コーパス（LDC2007T38, 2001 年 1 月～2004 年 12 月、単語数は 6 千万以上）である。

3.2.4.2 実験

実験は以下の手順で行った。

(1) コーパスを構文解析する。

文脈情報は N-gram と構文情報の二種類を用いた。予備実験では、構文情報を利用した方が良い結果がでたので、今回の実験には構文情報を素性として用いた。Stanford の中国語パーザーを用いて構文解析を行った。

(2) 構文解析の結果から、候補単語と素性を抽出する。

中国語パーザーの出力は全てをそのまま使えるほど精度が高いわけではないため、高い精度で解析されていると期待される解析結果を部分的に抽出し同義語抽出に用いた。全ての関係の中で、Dobj 関係の精度が一番高いため、候補単語と素性は Dobj 関係から生成した。頻度 30 以上の単語を候補単語として抽出し、それらに対する素性ベクトルを生成した。

(3) 素性選択基準を利用して、素性の数を削減する。

教師あり手法にとって、素性の数は最も重要なことである。素性の数が多すぎると、計算時間

および使用するメモリの点から学習が非常に困難となる。素性の選択には、二つの基準、頻度と文脈の重要度を用いた。まず、頻度 30 以上の素性を有効な素性として抽出した。次に、多くの単語が利用する文脈は最も重要な文脈という文脈の重要度基準で選択した。文脈の重要度は次の式で計算する。

$$df(c) = \left| \{ w \mid N(w,c) > 0 \} \right|$$

ただし、wは候補単語で、cは素性で、$N(w,c)$ は w と cの共起頻度である。重要度が 30 以上の素性は有効な素性として抽出した。選択する前の素性の数は 64,045 で、選択後 1,719 になった。

(4) 類似度を計算する。

類似度は次のアルゴリズムで計算する。まず、汎用同義語辞書に載っている素性対については、その類似度を 1 とする。汎用同義語辞書に載っていない素性対については、それらの類似度を計算し、ある閾値以上の類似度を有効類似度とする。次に、有効類似度を用いて候補単語間の類似度を計算する。得られた候補単語間の類似度を用いて、素性間の類似度を再度計算して、候補単語間の類似度も再度計算する。この計算を繰り返して、候補単語全体に対する類似度の差がある値以下となったら、計算を終了する。

汎用同義語辞書については、1984 年に発表された「同義語詞林」(Mei Gia-Chu et al. 1984) を汎用同義語辞書として用いた。

```
Input W:{wi | i=1...N },F:{ fj | j=1...M},S{s,set of similarity},θ: end threshold
          δ : end criteria;
0         start:
1         δ=0
2         for j=1 to M do
3                   for l=j+1 to M do
4                             ComputeSim(fj,fl)
5                             δ=δ+| sjl- ComputeSim(fj,fl) |
6                             Add ComputeSim(fj,fl) to S
7                   end for
8         end for
9         if δ< θ
10                  go to end
11        end if
12δ=0
13        for i=1 to N do
14                  for m = 1 to N do
```

```
15                          ComputeSim(wi,wm)
16                          δ=δ+|sjm- ComputeSim(wj,wm) |
17                          Add ComputeSim(wj,wm) to S
18                  end for
19          end for
20if δ< θ
21                  go to end
22end if
23end
```

3.2.4.2　実験の結果

平均精度（Mean Mean Average Precision）、RKL(Average Rank of Last Synonym L)とTop1 の三つの尺度を用いて実験結果を評価した。

表 1 は今回の比較実験の結果である。有効素性を選択するために、閾値を「0.4, 0.5, 0.6, 0.7, 0.8」に設定した。表からわかるように、閾値 0.6 以上の素性間類似度を使うと、同義語の抽出精度が上がることがわかる。閾値 0.7 と閾値 0.8 以上の類似度では、同じ抽出精度であることがわかる。

表 1　実験の結果

	閾値	MAP	RKL(1181)	TOP1
従来の手法		0.06958	1277.98182	0.12091
提案手法	0.4	0.05192	1389.36584	0.09652
	0.5	0.06165	1530.10885	0.10982
	0.6	0.06994	1298.94144	0.12110
	0.7	0.07327	1008.55741	0.14729
	0.8	0.07327	1008.55741	0.14729

3.2.5　まとめ今後の課題

本稿では、素性間の類似度による同義語抽出への影響に関する調査を行った。素性間の類似度を用いることにより、同義語抽出の精度が向上したが、まだ実用に供する精度には至っていない。それについてはいくつかの原因が考えられる。まず、中国語の構文解析の精度が低いことが原因の一つと考えられる。他の原因としては分布類似度により抽出された単語対が同義語ではなく、類義語が多く抽出された可能性が考えられる。今後は、実用に用いられうる中国語の同義語抽出に向けて研究を行いたい。

参考文献

Dekang Lin. Automatic retrieval and clustering of similar words. In Proceedings of the

COLLING/ACL 1998, pages 786-774, 1998.

J.R. Curran and M. Moens. Improvements in automatic thesaurus extraction. In *Proceedings of the Workshop on Unsupervised Lexical Acquisition*, pages 59-67, 2002.

Julie Elizabeth Weeds. Measures and Application of Lexical Distributional Similarity, Ph.D. thesis, University of Sussex, 2003.

Mei, Gia-Chu. 1984. Cilin- Thesaurus of Chinese words. (in Chinese) Hong Kong.

Zelling Harris. Distributional Structure. Katz, J.(ed.) *The Philosophy of Linguistics. Oxford University Press,* pages 26-47, 1985.

３．３　構文的共起と意味クラスを用いた訳語選択

静岡大学　綱川 隆司
梶 博行

3.3.1 はじめに

本稿では，訳語選択の知識をコンパラブルコーパスから獲得する手法を提案し，動詞の訳語選択を行うために構文構造と意味クラスを考慮した関連語抽出手法について述べる．

機械翻訳を行う際には原文に含まれる各単語・語句に対して適切な訳を選ぶ必要がある．多くの単語には複数の訳し方が存在し，もとの単語が多義語の場合は原文の意図するものと同じ意味を持つ訳語を選択しなければ誤った訳文となる．このような訳語の選択において手掛かりとして考えられるもののうち，本稿では周辺に現れる語に着目し，コーパスから得られる周辺単語の出現頻度から周辺単語とその訳語の関連性を表す関連語－訳語関連行列を求める．ここで用いるコーパスとしては文単位で対訳関係が付与されている二言語パラレルコーパスが考えられるが，利用できる分野や言語対が限られる．そこで，文レベルでは対訳でないが内容的に概ね対応した文書の集合や同一分野の各言語の単言語コーパスを組み合わせたコンパラブルコーパスを用いる手法を提案する．対訳語の対をコンパラブルコーパスから獲得する方法は比較的古くから研究されているが，本研究では訳語選択の知識をコンパラブルコーパスから獲得する手法の開発を目的とした．

提案方法の基本アイデアはコンパラブルコーパスを用いた教師なしの語義曖昧性解消手法として提案したものである (Kaji and Morimoto, 2002)．多義語の語義はそれぞれ同義の訳語の集合で表されるとし，それぞれの言語のコーパスから共起に基づく相関が高い語の組を関連語ペアとして抽出し，対訳辞書を介してアラインメントをとることにより，多義語の関連語が多義語の語義を決定する手がかりが得られる．例えば，英語コーパスと日本語コーパスからそれぞれ(tank, soldier), (戦車，兵士)なる関連語ペアが抽出され，これらが対応づけられることにより，多義語 tank の関連語 soldier が「軍事用車両」としての語義を支持することがわかる．コンパラブルコーパスでは対応する関連語ペアがそろって抽出されるとは限らず，また対訳辞書を介した機械的なアラインメントでは誤った対応が得られることも多い．この問題を解決するため，「互いに関連する関連語は同じ語義を支持する」という仮説に基づき再帰的に定義される関連語－訳語関連度を反復計算するアルゴリズムを提案した．例えば，tank の関連語 troop（軍隊）に対しては，日本語側で(戦車，軍隊)といった関連語ペアがコーパスに出現しない場合でも，soldier と troop が互いに関連するならば troop は「軍事用車両」としての語義を支持すると考えることができる．訳語選択においては，上記方法において各訳語候補がそれぞれ一つ語義であるとした特殊な場合と考えることができる．

また，上記方法では名詞の語義曖昧性対象を対象としているが，動詞の訳語選択に応用するには仮説を修正する必要がある．例えば，動詞 hit の訳について，相互に関連のある目的語 factory と ball はそれぞれ hit の異なる訳「（工場を）攻撃する」「（ボールを）打つ」を支持すると考えられる．そこで，相互に関連のある関連語の代わりに，『同じ意味クラスに属する名詞は同じ訳語を

支持する』という新たな仮説を用いる．この場合，同じ意味クラスに属する factory, plant, facility 等に対しては同一の訳語「攻撃する」を支持すると考えられる．

3.3.2　関連語－訳語関連行列に基づく訳語選択

3.3.2.1　共起頻度に基づく指標の計算

訳語選択の手掛かりとなる関連語を求めるための共起頻度に基づく指標に Dice 係数 (Smadja, 1993) を用いる．コーパスに含まれる動詞－目的語ペアおよび主語－動詞ペアについて，以下のパターンを用いて構文共起を抽出し，共起頻度を求める．

【英語】

Otero (2008) による品詞の正規表現によるパターンマッチングを用いて抽出できる動詞－目的語ペアおよび主語－動詞ペアを構文共起とする．用いた抽出パターンを正規表現で記述すると以下の通りである．

- 動詞－目的語ペア：(動詞)(限定詞｜副詞)*(名詞)
- 主語－動詞ペア：(限定詞)*(名詞)(副詞)*(動詞)

【日本語】

文節列によるパターンマッチングを用いて抽出できる動詞－目的語ペアおよび主語－動詞ペアを構文共起とする．以下の２つのパターンを用いて抽出する．

- 動詞とその前方にある n 個の名詞＋格助詞からなる文節
- 動詞（連体形）とその直後にある名詞

動詞 x と名詞 y の出現回数をそれぞれn_1, n_2，構文共起回数を m とする． このとき，x と y の Dice 係数 $\mathrm{Dice}(x,y)$ を以下の式で求める．

$$\mathrm{Dice}(x,y) = \frac{2m}{n_1 + n_2}$$

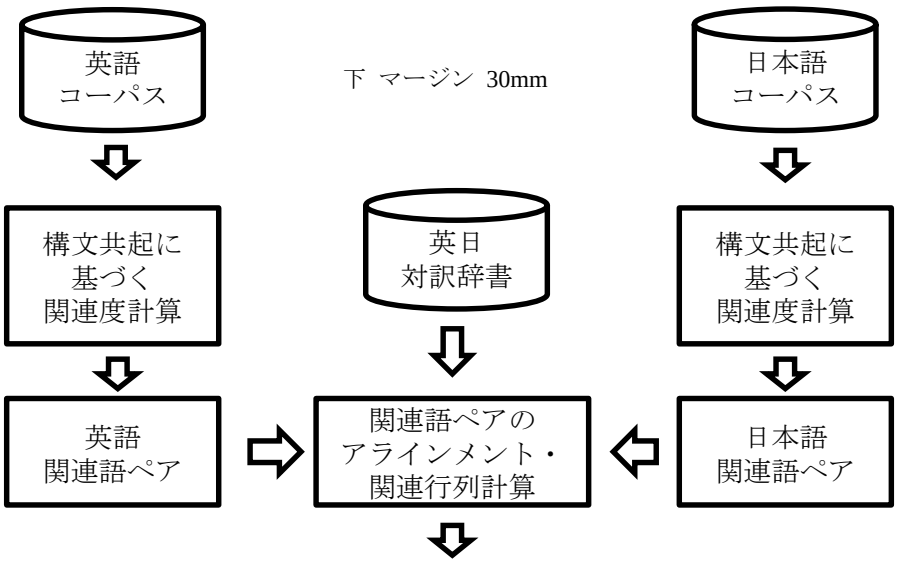
3.3.2.2　関連語－訳語関連行列

図 1 に関連語－訳語関連行列の計算手法を示す．

まず入力言語および出力言語それぞれの単言語コーパスに含まれる名詞を列挙し，全ての名詞対について共起頻度に基づく指標 $\mathrm{Dice}(f'(i), f)$を計算する．各名詞について，指標が高い語から順に関連語として抽出する．

次に，関連語－訳語関連行列を以下のように計算する．「同じ意味クラスに属する名詞は同じ訳語を支持する」という仮説に基づき，対象語 f の第 i 関連語$f'(i)$ と，f の j 番目の訳語$e(j)$ の関連度$C_f(f'(i), e(j))$を以下の反復計算で得る．

$$C_f^{(n)}(f'(i), e(j)) = \mathrm{Dice}(f'(i), f) \times \frac{\sum_{f'' \in A(f, f'(i))} \mathrm{Sim}(f'(i), f'') \cdot C_f^{(n-1)}(f'', e(j))}{\max_k \sum_{f'' \in A(f, f'(i))} \mathrm{Sim}(f'(i), f'') \cdot C_f^{(n-1)}(f'', e(k))}. \quad (1)$$



関連語－訳語関連行列

<lose>	失う	倒れる	敗れる	落とす
game	0.00180	0.0000569	**0.00365**	0.00309
life	0.000460	0.0000115	0.0000113	**0.000737**
right	**0.000291**	0.0000060	0.0000020	0.0000020

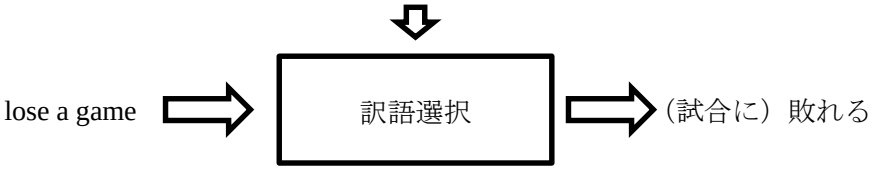


図 1: 関連語－訳語関連行列の計算

ただし，*n* は反復計算のサイクル，$A(f, f'(i))$は対象語 *f* と関連語 *f'*(*i*)に共通の関連語の集合である．すなわち，$A(f, f'(i)) = \{f'' | \mathrm{Dice}(f, f'') \geq \theta, \mathrm{Dice}(f'(i), f'') \geq \theta\}$．また，$\mathrm{Sim}(f'(i), f'')$ は$f'(i)$とf''の WordNet から求めた類似度とし，以下の式で定義する．

$$\mathrm{Sim}(f'(i), f'') = \frac{1}{\mathrm{shortest}(f'(i), f'') + 1}$$

ただし，$\mathrm{shortest}(f'(i), f'')$ は WordNet における上位下位関係からなる階層構造において，$f'(i)$からf''への最短パス長とする．$f'(i)$とf''が同一の同義語集合に属する場合は最短パス長を 0，また，$f'(i)$からf''へのパスが存在しない場合は $\mathrm{Sim}(f'(i), f'') = 0$ とする．

反復計算の初期値は以下の式で求める．

$$C_f^{(0)}(f'(i), e(j)) = \begin{cases} \dfrac{a(f'(i), e(j))}{\sum_k a(f'(i), e(k))} & (\Sigma_k a(f'(i), e(k)) \neq 0), \\ 0 & (\text{それ以外}) \end{cases}$$

$$a(f'(i), e(j)) = \begin{cases} 1 & (\exists e'.\ (f, f'(i)) \approx (e(j), e')) \\ 0 & (\text{それ以外}) \end{cases}$$

ここで，$(f, f'(i)) \approx (e(j), e')$は，$f$ と $e(j)$，および $f'(i)$と e'がそれぞれ対訳辞書に訳語対として存在することを示す．

以上から，曖昧性なく対訳関係にある関連語ペアの組が種となって，関連語と訳語の間の関連度が反復計算される．

3.3.2.3　関連語－訳語関連行列を用いた動詞の訳語選択

訳語選択は以下のようにして行う．ある文に含まれる動詞f_iに対して訳語 $e(1), e(2), \ldots, e(J)$があるとき，訳語 $e(j)$に対するスコアを以下の式で計算する．

$$\mathrm{Score}(f_i, e(j)) = C_f(Obj(f_i), e(j)). \quad (2)$$

ただし，動詞 f_i の目的語 を$Obj(f_i)$ とする．また，ある$f_i, e(j)$の組について$C_f(f_i, e(j))$の値が求められていない場合は 0 とする．

訳語のうち，最もスコアが高い訳語を選択する．また，全ての訳語についてスコアが 0 の場合は出力なしとする．

3.3.3　実験

英語および日本語のコーパスから関連語－訳語関連行列を各手法により求め，英文に含まれる各動詞の訳語を出力して評価する実験を行った．

関連行列を計算するためのコーパスには, English Gigaword コーパスの New York Times（2004 年, 277MB）および毎日新聞コーパス（2004 年,140MB(UTF-8)）を用いた．関連行列の反復計算に用いる対訳辞書には EDR 電子化辞書の英日・日英対訳辞書，EDICT (Breen, 1995)および英辞郎を組み合わせたものを用いた．予備実験として New York Times（2005 年 1 月）を対象に，含まれる動詞に対して訳語の出力を行った．

図 2，3 に正しい訳が得られると考えられる例を示す．図 2 の動詞 lose に対する関連行列は，目的語 game に対して訳語「(試合に）敗れる」，life に対して「(命を）落とす」，right に対して「(権利を）失う」を支持することを示しており，評価対象の文に現れる動詞 lose とその目的語 rights から，訳語を正しく選択することができると考えられる．この文では，名詞の訳語選択手法のように目的語に限らず周辺に現れる訳語を考慮すると Astros といった野球チームを表す固有名詞によって「敗戦する」という訳語が選択されてしまうが，目的語に限って考慮することで広い文脈に左右されず正しい訳語を選択できる．

同様に，図 3 の動詞 discipline に対する関連行列は athlete に対して「鍛える」，officer に対して「罰する」, student に対して「まとめる」を支持しており, 評価対象の文に現れる動詞 discipline の訳をその目的語 athlete から適切に選択できるものと考えられる．

一方，図 4 の動詞 remember の訳「覚える」「思い出す」については，この文の例では目的語 winter に対して適切な「覚える」が選択されていると考えられるが，実際には文脈によってどちらの訳も選択される可能性があるため，目的語だけで適切な訳を選択するのが困難なケース

The Houston Astros, who traded for Beltran last summer, must sign him by Jan. 8 or **lose** negotiating *rights* with him until May 1.

<lose>	失う	倒れる	敗れる	落とす	…
Astros	0.000100	**0.000125**	2.95×10^{-5}	5.35×10^{-5}	
game	0.00180	5.69×10^{-5}	**0.00365**	0.00309	
life	0.000460	1.15×10^{-5}	0.0000113	**0.000737**	
right	**0.000291**	6.00×10^{-6}	2.00×10^{-6}	2.00×10^{-6}	
…					

図2: 動詞loseの訳語選択例

He is a tremendously **disciplined** *athlete*, a very focused athlete, humble with his teammates and respectful of his skills,

<discipline>	まとめる	鍛える	罰する	…
athlete	1.16×10^{-5}	$\mathbf{3.60 \times 10^{-5}}$	9.52×10^{-9}	
officer	5.51×10^{-5}	5.61×10^{-7}	**0.000153**	
student	$\mathbf{6.00 \times 10^{-5}}$	5.76×10^{-6}	2.68×10^{-8}	
…				

図3: 動詞disciplineの訳語選択例

But for all the high-profile pitchers in demand this off-season, the *winter* of 2004-5 may be **remembered** for one powerful hitter.

<remember>	覚える	思い出す	思い浮かべる	…
moment	0.000254	**0.000390**	1.99×10^{-8}	
people	**0.000286**	0.000185	5.87×10^{-5}	
winter	$\mathbf{4.30 \times 10^{-5}}$	3.29×10^{-5}	2.87×10^{-6}	
…				

図4: 動詞rememberの訳語選択例

と思われる．また，「思い浮かべる」等のような頻度の低い訳語に対する値がほとんどの関連語に対して非常に小さくなりほぼ選択されなくなる問題がある．

3.3.4 関連研究

単言語における語義曖昧性解消は，辞書やコーパス等のデータを使って教師なし学習を行う手法が提案されてきた (Ide and Veronis, 1998)．品詞等の文法的情報，構文的関係にある語，お

よび周辺に共起する分野に関する語を曖昧性解消の手掛かりとして用いている．Dagan and Itai (1994) は翻訳先言語の単言語コーパスを用いた語義曖昧性解消手法を提案している．本研究では単言語ではなく二言語コンパラブルコーパスを用いることによって二言語間の関係を考慮することができる．

Li and Li (2002) は単語の翻訳の曖昧性解消を対訳辞書の対応付けをもとにブートストラッピングによって分類器を構築し行っている．本研究では辞書の対応関係は反復計算の種として用いコーパスから求めた手掛かり語を考慮に加えている．

Vickrey et al. (2005) は統計的機械翻訳に文脈を考慮した訳語選択を素性として導入し，訳語の選択を試みている．文全体の統計的機械翻訳システムへの導入が大きな課題である．

2.3.5 おわりに

本稿ではコンパラブルコーパスから構築した関連語－訳語関連行列によって動詞の訳語選択を行う手法を示し，実際に関連行列を求めて訳語選択に対するフィージビリティを確認した．今後の課題として，句動詞等の連語により意味が変化または限定される場合の考慮，同義訳語のクラスタリング等の訳語の整理による精度向上，動詞に係る目的語・主語・前置詞句内の語などの種類による区別等が考えられる．

参考文献

Breen, J.W. 1995. Building an Electronic Japanese- English Dictionary. In *Proc. of the Japanese Studies Association of Australia Conference*.

Dagan, Ido and Alon Itai. 1994. Word Sense Disambiguation Using a Second Language Monolingual Corpus. *Computational Linguistics*, 20(4):563-596.

Ide, Nancy and Jean Veronis. 1998. Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):1-40.

Kaji, Hiroyuki and Yasutsugu Morimoto. 2002. Unsupervised Word Sense Disambiguation Using Bilingual Comparable Corpora. In *Proc. of the 19th International Conference on Computational Linguistics*, pages 411-417.

Li, Cong and Hang Li. 2002. Word translation disambiguation using bilingual bootstrapping. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 343-351.

Otero, Pablo Gammallo. 2008. Evaluating Two Different Methods for the Task of Extracting Bilingual Lexicons from Comparable Corpora. In *Proc. of LREC 2008 Workshop on Comparable Corpora*, pages 19-26.

Smadja, Frank. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143-177.

Vickrey, David, Luke Biewald, Marc Teyssier and Daphne Koller. 2005. Word-sense disambiguation for machine translation. In *Proc. of the Conference on HLT/EMNLP*, pages 771-778.

３．４　特許文ツリーバンキングのための文法枠組みに関する考察

東京大学　　王 向莉
辻井 潤一

3.4.1　はじめに

ツリーバンクは自然言語処理及び言語学研究のための重要な資源である。ツリーバンクはある文法枠組みに基づいてテキストに統語構造を付与することで作成される。そのため、選択された文法枠組みはツリーバンクから得られる文法情報の種類を決めるだけでなく、ツリーバンクの構築の効率と作成されたツリーバンクの品質、およびツリーバンクを構築する際の方法論にも深くかかわっている。特許文には、多くの分野の専門用語や独特の語彙・言い回しが含まれている。しかし、特許文ツリーバンクはまだ存在していない現状である。本稿では、いくつかの代表的な文法枠組みについて、表示される情報の種類、アノテーターにとっての表示の直観性などの特性に着目しながら整理し、それらの特性が各枠組みによるツリーバンキングに与える影響について論じる。最後に、文法情報の豊かで、かつアノテーターにとって表示が分かりやすい文法枠組みをインターフェスにし、特許文ツリーバンキングすると同時に、裏側でほかの文法枠組み（例えば、HPSG）の構文木に変換することより、複数の枠組みの特許文ツリーバンクを同時的に構築する構想を示す。

3.4.2　文法枠組みごとの整理

3.4.2.1　依存文法 DG

DG では、図 1 に示すように、統語構造がある単語とその従属部の関係として定義され、句ノードのような情報がない。PDT（J. Hajic et al., 2000）は依存文法に基づいて作成された典型的なツリーバンクである。PDT は形態素情報、構文構造情報、意味構造情報の三つの段階でアノテートされた。

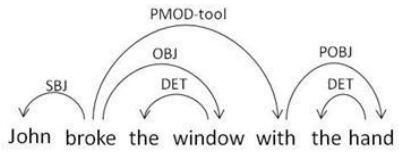


図１：依存文法に基づく構文木

3.4.2.2　句構造流文法

ここでの句構造流文法はチョムスキーの提案およびそれに基づいて発展してきたすべての枠組みを指す。本稿では、特に、文脈自由句構造文法および語彙化文法の一種である HPSG の２つについて考える。

3.4.2.2.1　文脈自由句構造文法 CF-PSG

CF-PSG はツリーバンキングのためによく選ばれる文法枠組みである。Penn Treebank (Marth et al., 2005) は典型的な CF-PSG に基づくツリーバンクである。CF-PSG は 図 2 に示すように文 1a を句構造で解釈する。CF-PSG に基づくツリーバンクでは、意味構造を直接表示しない場合が多い。例えば、1a と 1b は同じ意味構造を持つと考えられるが、それぞれ文の句構造による表示では、この意味の同一性は直接表示されない。また、句構造による表示では、文 1c と文 1d のように構文木上の述

語－項の位置関係と意味役割が 1 対 1 に対応しない場合に意味構造の同一性を直接読み取るのは困難である。

1a. John broke the window with the hand at school yesterday

1b. the window was broken by John with the hand at school yesterday

1c. John broke the window

1d. the window broke

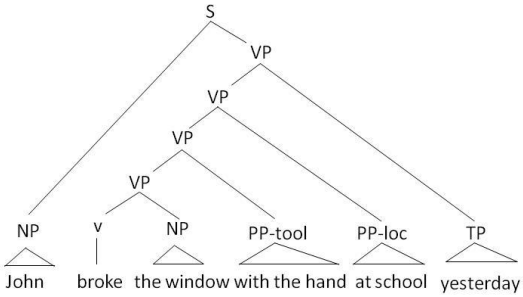


図 2：句構造文法に基づく構文木

3.4.2.2.2　主辞駆動句構造文法 HPSG

語彙化文法の一種である主辞駆動句構造文法 HPSG(Pollard and Sag, 1994) は PSG の拡張であり、CF-PSG と同じように文を句構造で解釈するが、構文木の各ノードに置かれるデータ構造（Sign）の中で、意味構造が直接表示される。この反面、Sign による表示は一般に複雑なものとなり、表示されている統語・意味構造をアノテーターが直観的に把握することは難しいと考えられる。

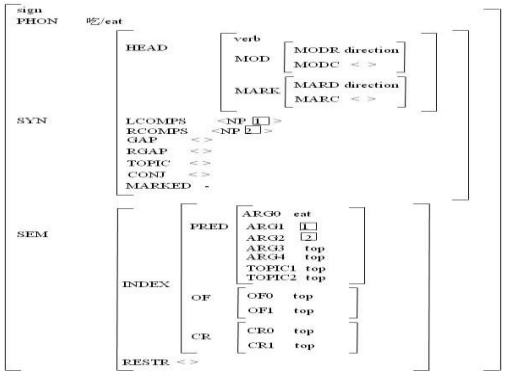


図 3：HPSG の語彙項目の例

3.4.2.3　文構造文法

文構造文法 SSG（王,宮崎 2007）は依存文法と句構造流文法の特性を併せ持つ文法枠組みである。CF-PSG と比べると、基本的な区別が二つある。

1) 文を細かな句構造で解釈するのでなく、文を述語と述語を中心とした構文要素からなる文構造で解釈する。具体的には、各述語に対する必須の構文要素および付加的な構文要素を、述語とともに構文木上の 1 つのレベルにまとめて表示する。
2) 文の意味構造を、述語と述語の周囲の構文要素との意味的依存関係として構文木上で直接表示する。

文 1a を例にして、SSG はどのように文を解釈するかについて説明する。文の述語は”broke”であり、その前の名詞句”John”が主語であるので、”Sn”で表示する。名詞句”the window”は目的語であるた

め、”On”で表示する。前置詞句”with the hand”は道具で、 “at school”は場所で、 時間詞句“yesterday”は時間の要素であるため、それぞれ PP-tool、PP-loc と TP で表す。すべての要素が図 3 に示すように文構造規則 1)に記述する。

図 4 と図 5 に示すように、文 3a と文 3b はそれぞれ、規則１）と規則２)で解析する。2 つの文のどれにあっても、”John”は意味上の主語であり、”the window”は意味上の目的語である。

規則 1) s → Sn V On PP-tool PP-loc TP

規則 2) s → On BE V BY Sn PP-tool PP-loc TP

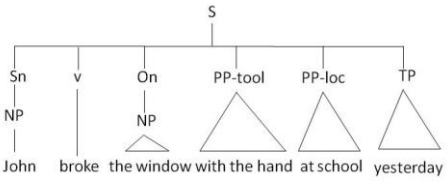


図 4：文構造文法に基づく構文木(1)

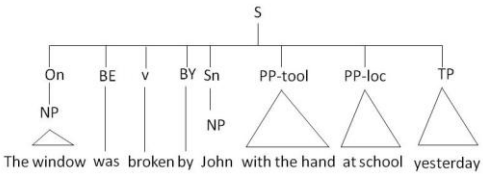


図 5：文構造文法に基づく構文木(2)

SSG では、文 2a のように複数の述部を持つ並列構造を扱うのが難しい場合がある。これは、主語位置の NP がそれぞれの節に対して異なる意味役割（意味上の主語と目的語）を持つため、主語位置の NP に対応するノードのラベルとしてそれらを表示することができないためである。

このような場合、図 6 のように、主語が必要である節に CL_Sn_gap、目的語が必要となる節に CL_On_gap というラベルを与えることで意味・統語構造を両方表示するが、図 4 のような単純な場合と異なる取り扱いとなり、また、意味的な依存関係を表示から直観的に読み取ることは難しいという問題がある。ただし、現実的にはこのような構造が現れる頻度はそれほど高くないと考えられる。

2a. John slept on road and was robbed.

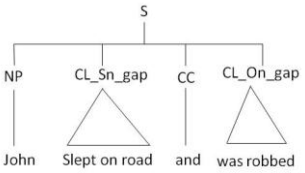


図 6：SSG の節の処理

3.4.3 文法枠組みの比較

3.4.3.1 文法情報の豊かさと直観性の比較

図 1 に示すように、DG は述語とそれに直接依存する従属部の意味上の関係を直観的に分かりやすく表示できるが、句構造の情報が欠けている。そのため、例えば単純な名詞句などであっても、構文要素のかたまりが直観的に把握しにくい。

これに対し、PSG では句構造を利用することで、スコープなどの情報を表示できる場合がある。また、PSG に基づくコーパスである Penn Treebank では、句構造に加えて -SBJ、-OBJ といった文法機能タグを句ラベルに付加する形でアノテートしている。しかし、述語・項関係のように、構文木上

の位置関係と必ずしも一対一に対応しないような情報をアノテートする場合、句ラベルをさらに拡張することで表示する方法は（原理上は可能であるにせよ）アノテーターにとって見やすい表示であるとは言えないだろう。HPSG のような語彙化文法は PSG で表示されるような構文構造に加え、述語項構造のような意味情報を表示するためのシステムを含んでいる。しかし、既に図 3 に示したように統語情報と意味情報、さらに両者の関係を同時に含む表示は非常に複雑になる場合があり、これを直接アノテーターに提示するための表現形とするのは難しい。

SSG は構文構造情報と意味構造情報を分けずに 1 つの文構造規則で表示するため、図 4 と図 5 に示すように、意味上の主語 Sn と意味上の目的語 On のような述語と項、および修飾句との意味関係を構文木上で直接表示できる。また、図 5 に示したような場合を除けば、構文木上の一つのレベルに述語とそれに依存する句が並ぶため、CF-PSG や HPSG の意味表示のような複雑な記法を必要としない。また、ある程度まで句構造の情報を表示するため、特に頻度の多い名詞句などは構文要素としてのかたまりを直観的に把握できる。さらに、空範疇と co-indexing の仕組みを導入することで、HPSG では解析が難しい、文 3a のような例を図 7 のように述語項関係が見やすい形で表示することも可能である。

3a. John likes apples and Mary oranges.

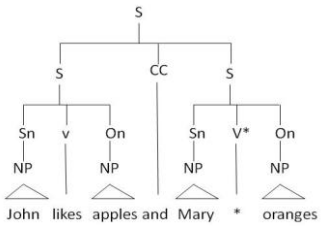


図 7：SSG でアノテートしやすい文例

3.4.3.2 方法論の比較

文法枠組みの文法情報の豊かさと直観性がツリーバンキングの方法論にかかわっている。従来のツリーバンクは静的な方法と動的な方法のどちらかに従って作られたものである。

静的方法論とは、複数のアノテーターが自分の言語的な直感に頼って、1 つずつの構文木をアノテートするという方法である。従来の DG および PSG に基づくツリーバンクは静的な方法に従って構築されるものが多い。

動的な方法とは、あらかじめ文法規則を用意しておき、その文法規則にしたがって解析した結果を文に付与する方法である。LinGo Redwoods(Stephan et al.,2002)は動的な方法に従って、語彙化文法の一種である主辞駆動句構造文法HPSGに基づいて構築されたツリーバンクである。

DG や PSG のような文法が比較的乏しい文法枠組みに基づいてツリーバンキングをする場合、広範囲の文を被覆するためには過剰生成する文法を使わざるを得ない場合がしばしばあるため、動的な方法に向いていない。

HPSG のような詳細な文法的制約を表現できる文法は、DG や PSG よりも、動的な方法に向いている。その一方、HPSG のような語彙化文法は、その統語・意味構造の表示が非常に複雑なため、文法をあらかじめ用意することなしで、アノテーターが一文ずつアノテートするのは非常に難しい。そのため、語彙化文法は静的な方法に向いていないと言える。

SSGはDGと同様に表示の直観性に優れ、PSGと同程度の単純な形式をもつため、静的方法によるアノ

テーションが可能だと考えられる。また、各文法規則を述語によって語彙化することで語彙化文法と同様の詳細な制約が記述できるため、動的方法論でツリーバンキングすることも考えられる。

3.4.4　新しい方法論

統語構造・意味構造がともに表示でき、かつ直観性のよい SSG をインターフェスにし、SSG に基づいてツリーバンキングをするのと並行し、SSG の豊かの文法情報を使って、アノテートされた SSG 構文木を、ほかの文法枠組み（例えば一般性に優れた文法を記述可能であるが、表示が複雑である HPSG のような語彙化文法）における構文木に変換し、同時に複数の枠組みでツリーバンクを構築する方法が考えられる（図８）。

図 8：同時的に複数のツリーバンクを構築する構想

3.4.5　結論と展望

本稿では、ツリーバンキングという側面から、依存文法 DG、PSG 流文法（CF-PSG、HPSG）および文構造文法 SSG を比較した。ツリーバンキングする際、文法枠組みの直観性や文法情報の豊かさが望まれていることを検討した。文構造文法 SSG は直観性に優れたかつ豊かな文法情報を持つため、ツリーバンキングするのに向いている文法枠組みであることが分かった。また、SSG をインターフェスにして、同時に複数の文法枠組みのツリーバンキングする方法が提案された。将来的には、提案された方法に基づいて、中国語、日本語、英語などの特許文ツリーバンクの効率的な構築が期待される。

参考文献

Martha Palmer and Daniel Gildea and Paul Kngsbury (2005). *The Proposition Bank: An Annotated Corpus of Semantic Role*. In Computational Linguistics. Vol. 31 Issue 1, March 2005.

Stephan Oepen, Dan Flickinger, Kristina Toutanova, Christoper D. Manning.(2002). LinGo Redwords: A Rich and Dynamic Treebank for HPSG. In Proc. TLT 2002.

Mitchell P. Marcus, B. Santorini and Mary Ann Marcinkiewicz (1994).*Building A Large Annotated Corpus of English: The Penn Treebank.* Computational Linguistics, Vol. 19, No. 2. (1994), pp. 313-330.

Bond F., S. Fujita, C. Hashimoto, D. Kasahara, S. Nariyama, E. Nichols, A. Ohtani, T. Tanaka, S. Amano (2004).*The Hinoki Treebank: Working Toard Text Understanding.* In LINC-04.

Carl Pollard, Ivan A. Sag (1994). *Head-Driven Phrase Structure Grammar.* Chicago: University of Chicago Press.

王向莉, 宮崎正弘(2007). *文構造文法に基づく中国語構文解析*. 自然言語処理, vol.14 no.2, pp.69-93.

Jan Hajic, Alena Bohmova, Eva Hajicova, Barbora Vidova Hladka (2000). The Prague Dependency Treebank: A *Three-Level Annotation Scenario*. In A. Abeillé (ed.): *Treebanks: Building and Using Parsed Corpora*, Amsterdam:Kluwer, 2000, pp. 103-127.

Miyao, Yusuke (2006). *From Linguistic Theory to Syntactic Analysis: Corpus-Oriented Grammar Development and Feature Forest Model*. PHD Thesis.

４．　構造を持った定型表現の自動獲得と機械翻訳での利用

京都大学　望月 道章
中澤 敏明
黒橋 禎夫

4.1 はじめに

現在の機械翻訳では単語よりも大きなフレーズを単位として翻訳を行うことが一般的である。しかし、ここでのフレーズは対訳コーパスから自動的に推定された単語対応をヒューリスティックなルールを用いて拡張することで得られたものであり、言語的に意味のある句とは必ずしも一致しない。そのため、複数語で一つの意味を持つ定型表現を扱う際に問題が起こる。定型表現の多くは“方 が 良い”や“in order to”などのように一つ以上の機能語を伴っているが、これらの機能語は単体では意味を成さないか、他の語と結びついて異なる意味を表すので、対訳文中で出現した場合、相手言語文に一対一で対応する単語がないことがほとんどである。そのため、対応の推定を誤ってしまい、正しい翻訳知識が獲得されず、翻訳誤りを引き起こしてしまう。

対応を誤ってしまう原因は各単語を個別に扱っていることであり、これを解決するためには意味を持つ単位として正しい表現を考慮する方法が考えられ、既にいくつかの手法が提案されている[3, 1]。しかしこれらの研究では単語列上で連続した表現しか考慮しておらず、中国語の“在 ～ 中”(“～ において”) のような単語列上は連続しない定型表現を扱うことができない。そこで本研究では依存構造木から定型表現を自動的に獲得し、知識として機械翻訳で利用する手法を提案する。定型表現の獲得はその表現の出現頻度や周辺語の異なり数に基づいたスコアを用いて行う。依存構造木を用いることで単語列では不連続であっても、直接の依存関係が存在していれば定型表現として獲得することができる。また、自動獲得された定型表現を対訳文内の単語・句アライメントで利用することにより、アライメント精度の向上を目指す。

4.2 依存構造木からの定型表現の獲得

本手法では任意の表現に対してスコア付けを行い、その値が閾値以上の表現を定型表現として獲得する。定型表現はまとまりで頻繁に出現し、連接する単語の種類が多いと考えられる。この特徴を取り入れた指標としてC-value[2]を依存構造木に拡張したものを用いる。

4.2.1 C-value

C-valueとは単語列を対象としたコーパス中のコロケーションを判定するためのスコアであり、以下の式で定義される。

$$\text{C-value}(a) = \begin{cases} \log_2(|a|) \cdot f(a) & (T_a = \phi) \\ \log_2(|a|) \cdot \left(f(a) - \dfrac{\sum_{b \in T_a} f(b)}{\#(T_a)} \right) & (\text{otherwise}) \end{cases}$$

aは対象とする表現、$f(a)$と$|a|$はそれぞれaの頻度とaを構成する単語数である。T_aはaを内部に含むより大きな表現の集合であり、その異なり数を$\#(T_a)$とする。式の形から頻度($f(a)$)が高い表現であっても周辺の単語の種類($\#(T_a)$) が少なければ、値が小さくなることがわかる。

C-valueでは大きさに関係なくaを含む全ての表現をT_aとして扱っている。例えば、“in spite”のC-valueは“in spite of”、“increased in spite”、“in spite of the”などをT_aとして計算する。しかし、本研究では図1のようにaよりも一単語大きい表現だけをT_aとし、文頭側と文末側で別々にC-valueの計算を行い、両方が閾値以上の表現を獲得する。なお、図のA,B,C,X,Yは単語を表す。つまり、“in spite”の計算を行う際は“in spite of”と“increased in spite”のように文頭又は文末側に一単語が接続した表現だけをT_aとする。“in spite”はほとんどの場合“in spite of”に含まれて出現するため、文末側の値が低くなり、定型表現として獲得されず、“in spite of”のみが獲得される。

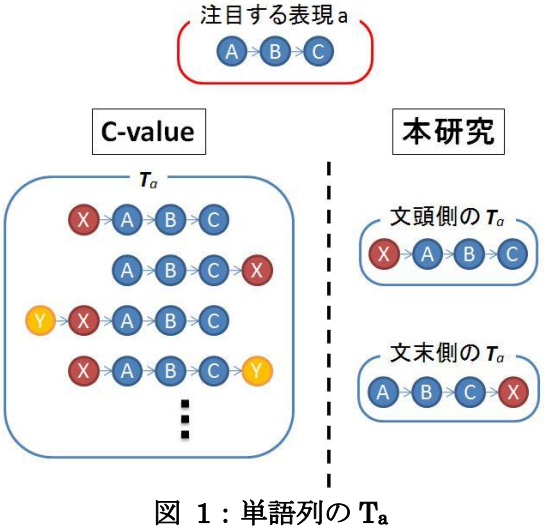
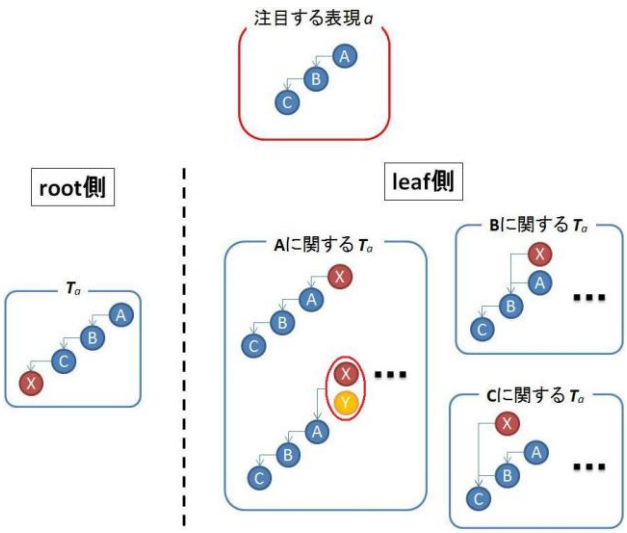


図 1：単語列の T_a

4.2.2 依存構造木からの獲得

依存構造木では文頭側と文末側の代わりにroot側とleaf側でT_aを区別し、計算を行う。root側に関しては係り先は一つしかないので単語列と同様に計算できるが、leaf側については単語列の場合と異なるため、T_aをさらに詳細に区別する必要がある。そこで本研究では以下の変更を加え、図2のようにT_aを区別した。なお、今後は依存構造木上の単語はノードと呼ぶ。



ノードごとにC-valueを計算

単語列の場合、単語は常に注目する表現の端のノードに接続していた。しかし依存構造木では全てのノードに単語が接続する可能性があり、同じ単語でも異なるノードに接続することが考えられる。このような異なりを全てT_aの一つとしてしまうと、あるノードに接続しやすい単語があっても他のノードに接続する単語の種類が多いため、#(T_a)が大きくなり誤って獲得されてしまう。

図 2：依存構造木の T_a

そこで、図2に示すように*a*のノードごとにT_aを区別しそれぞれについて計算を行い、その最小値をleaf側のスコアとする。これにより、あるノードに接続しやすい単語が存在すれば、そのノードのC-valueは小さくなるので、定型表現としては獲得されず、全てのノードについて接続する単語の種類が多い表現のみを獲得することができる。

同じノードに接続する単語をまとめて扱う

依存構造木では一つのノードに複数の単語が接続する場合があり、そのような表現をどの表現のT_aとして扱うかが問題になる。本研究では接続している単語をまとめて扱い、図2に示すように*a*の任意のノードに複数の単語が接続する表現もT_aとして計算を行う。また、同じ単語でも接続するノードに対し単語列上で前から接続している場合と後ろから接続している場合で区別して扱っている。

本研究のように依存構造木を利用した表現獲得の研究には葛原ら[7]やMartensら[4]の研究がある。葛原らは本研究と同じようにある表現のノードごとにスコアを計算することで獲得を行っている。しかし、英文作成を支援する表現の獲得が目的であり、節や句なども含んだ大きな表現も獲得されている。機械翻訳ではできるだけ小さい単位を扱う方が望ましいため、本研究とは獲得したい表現の大きさが異なる。また、Martensらはいくつかのスコアを用いて表現を獲得しているが、獲得した表現の具体的なアプリケーションへの応用は行っていない。

4.2.3 定型表現の獲得実験

提案手法による定型表現の獲得実験を行った。利用したコーパスは小規模論文コーパスと大規模Webコーパスである。論文コーパスは内山・井佐原らの方法により作成したJST日英抄録(約100万文対) [6]と日中科学論文(約70万文対)であり、各言語を単言語のコーパスとみなして獲得を行った。ただし、Webコーパスからの獲得を行ったのは日本語と英語のみである。次に、単語の区別であるが、単語は以下の情報が全て異なるものを一種類として扱った。

日本語: 代表表記, 品詞, 活用形

英語: 原形, 品詞 (“it”以外の代名詞, 三単現, 複数形は区別しない)

中国語: 表層語, 品詞

今回は獲得の対象を6単語以下の表現に限定した。また、獲得されたものの中には機械翻訳で扱うには不適切な表現があったので、以下に示す簡単なルールでフィルタリングを行った。

日本語: rootが名詞または格助詞

例: “システム が”、“を 解析”など

leafが“する”または名詞性接尾辞

例: “さ れて いる”、“性 に ついて”など

英語: be 動詞が動詞または形容詞と接続していない

例: “the system is”、“and ～ is”など

中国語: “进行”が動詞に接続していない

例: “对 ～ 进行”など

論文コーパスから獲得された定型表現の例をそれぞれ表1に示す。論文コーパスでは“こと ができる”や“in order to”など翻訳で有用な定型表現が獲得できていることが分かる。また、“在 ～ 中”や“as ～ as”など単語列上では不連続な定型表現も本手法により獲得できる。また、Webコーパスを用いた場合も“こと が できる”や“due to”などの定型表現が獲得されたが、論文コーパスの結果に比べると、“と いう”や“at least”などの一般的な定型表現が多く獲得されていた。

表 1：獲得された定型表現の例

日本語	英語	中国語
に ついて	this paper	在 ～ 中
で は	be carried out	是 ～ 的
本稿 で は	based on	的 方法
を 提案 する	in order to	一 个
こと が できた	this paper described	不 能
に ついて 述べる	due to	就 ～ 是
示唆 した	as ～ as	高 的
…	…	…

4.3 定型表現の利用

本研究ではベースラインシステムとして中澤らの用例ベース機械翻訳システム[23]を利用し、定型表現をアライメント時の制約として用いた。

4.3.1 ベースラインシステム

中澤らのモデルでは依存構造木上で統計的句アライメントを行っている。依存構造木を用いることで、言語構造が大きく異なる言語対でも柔軟に対応することができる。アライメント手法を簡単に説明すると、まず既存の統計的単語アライメントモデルにより単語レベルでの対応を推定し、これをヒューリスティックなルールで依存構造木上での句対応にマッピングする。これを初期状態とし、句対応確率と句の依存関係の確率を考慮してEMアルゴリズムにより繰り返しモデル推定を行う。EMアルゴリズムの途中により大きな句を獲得するステップがあることが特徴である。

4.3.2　定型表現による制限

機械翻訳では定型表現を構成する単語を個別に扱うと翻訳誤りの原因になってしまうため、定型表現はまとめて扱う必要がある。しかし、定型表現を構成する単語のアライメントが誤っているために依存構造木上で対応先が不連続になり、まとめて扱えない場合がある。その例を図3 に示す。ここでは、(■)が対応関係を表し、薄い青と濃い青の部分が正解の対応である。定型表現である“方 が 良い”が“方 ⇔ me”と“良い ⇔ should”という対応を持っているため、“方 が 良い”の対応先は依存構造木上で不連続になってしまう。そこで、本研究では定型表現はまとまりで一つの意味を持つので対応先でもまとまっていると考え、定型表現を構成する単語の対応先が依存構造木上で不連続になる対応を禁止するという制約を用いた。具体的には、定型表現を構成する単語の対応先が不連続になってしまう場合、対応確率が低い方を利用しない。こうすることで、図3 の“方 ⇔ me”がなくなり、“方 が 良い”をまとめて扱うことができる。

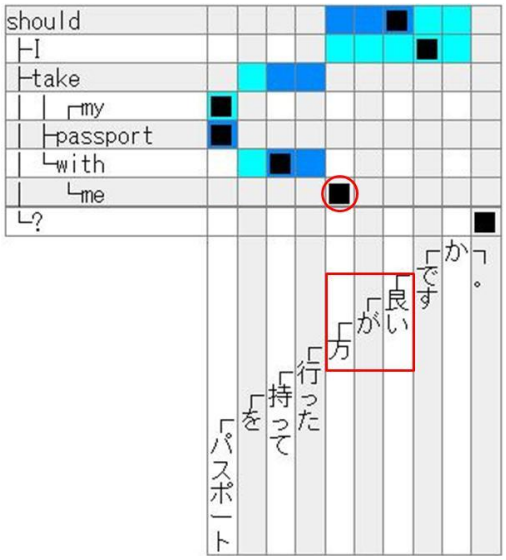


図 3：定型表 表 2：定型表現のスコア

定型表現	スコア
こと が できる	22830
が できる ます	22605
が できる	33572
ます か	20714

また、文内のどのまとまりを定型表現とするかも問題となる。本研究では前章で獲得した定型表現のうち文内に存在するものは基本的に全てを定型表現として採用する。ただし、候補がオーバーラップした場合は長い表現を優先する。もし、同じ長さであった場合はC-valueの高い方を採用する。この時のC-valueはroot側とleaf側の平均を取った値である。例えば、獲得された定型表現のC-valueが表2であった場合、“こと が できる ます か”という文では“こと が できる”と“ます か”の二つの定型表現が採用される。

4.4　アライメント実験

4.4.1　実験設定

定型表現を利用したアライメントを行い、精度への影響を調べた。定型表現は以下の4種類から獲得したものを利用し、ベースラインの結果と比較した。

- 論文コーパスの単語列
- 論文コーパスの依存構造木
- Webコーパスの単語列(日英のみ)
- Webコーパスの依存構造木(日英のみ)

実験には定型表現の獲得で利用した対訳コーパスを用いた。アライメントの評価には人手で正解を与えた日英480対訳文と日中500対訳文を利用し、以下の式で示されるPrecision、Recall、Alignment Error Rate(AER)を用いた。AERはアライメントの総合的な精度を示す指標であり、

その値が低い程精度が良い。

$$\text{Precision}=\frac{|A\cap P|}{|A|}\quad \text{Recall}=\frac{|A\cap S|}{|S|}\quad \text{AER}=1-\frac{|A\cap S|+|A\cap P|}{|A|+|S|}$$

Aがシステムの出力、PとSが正解である。S(Sure)は必ず必要な正解であり、図3の濃い青である。P(Possible)は英語の冠詞や日本語の助詞などのようにあっても誤りではない正解であり、図3の薄い青である。

4.5　実験結果

日英と日中の定型表現を利用したアライメントの精度をそれぞれ表3と表4に載せる。

結果を比較すると日英ではWebコーパスの単語列と依存構造木から獲得した定型表現を用いた場合、日中に関しては単語列から獲得したものを用いた場合の精度が最も良かった。この結果から定型表現がアライメントに有効であることが分かる。実際に改善した例を図4に載せる。この例では、定型表現“在 ～ 中”があるので、誤った“在 ⇔ は”の対応が禁止され、アライメントが改善している。

また、依存構造木を用いた場合、全体的な精度であるAERは単語列より低下しているが、Precisionが上昇していることがわかる。翻訳においてはPrecisionが高い方が正確な翻訳が行えるので、依存構造木から獲得した定型表現は翻訳に有効であると考えられる。

しかし、言語構造の違いなどによって定型表現の対応が正しくても対応先が不連続となってしまう場合があり、正しい対応が禁止され、Recall低下の原因となっている。定型表現の情報をどのようにアライメントの制約として利用するべきかについては今後さらに検討する必要がある。

表 3：日英アライメント精度

定型表現	Pre.	Rec.	AER
なし	84.53	65.53	**25.78**
単語列(論文)	85.10	65.18	25.80
依存構造木(論文)	**85.22**	65.08	25.81
単語列(Web)	84.90	**65.59**	**25.63**
依存構造木(Web)	85.13	86.46	**25.63**

表 4：日中アライメント精度

定型表現	Pre.	Rec.	AER
なし	86.49	**77.31**	18.09
単語列(論文)	86.74	77.25	**18.00**
依存構造木(論文)	**86.78**	77.17	18.02

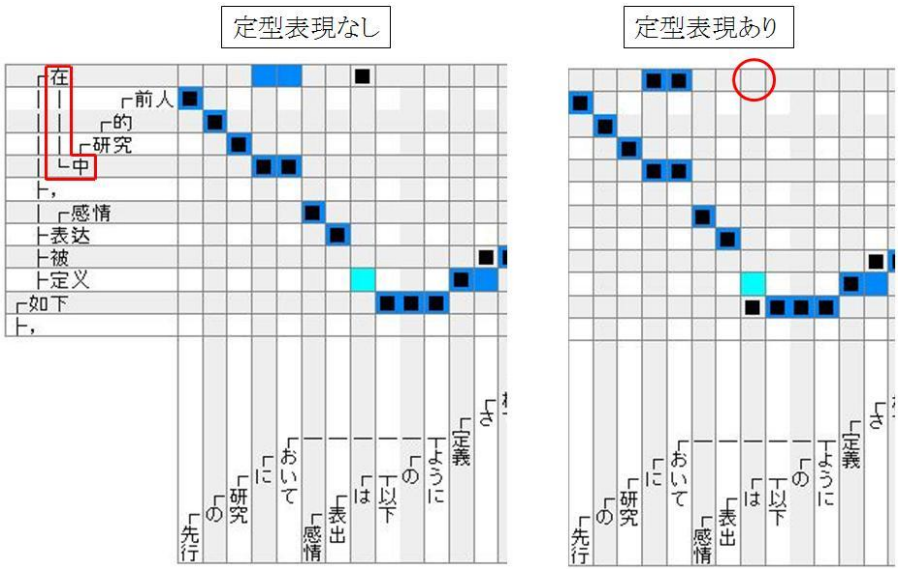


図 4：アライメント改善例

4.6 おわりに

本研究では定型表現をコーパスから自動的に獲得し、アライメント時に制約として利用する手法を提案した。定型表現はコロケーション獲得の指標であるC-valueを拡張することで依存構造木から獲得しており、単語列上では連続しない表現も獲得できる。また、アライメントでは定型表現を構成する単語の対応先が依存構造木上で不連続になる対応を禁止するという制約を用いた。実験は日英、日中間で行い、そのどちらでも定型表現を利用するとアライメントの精度が向上することを確認した。今後の課題は、制約により誤って禁止される問題を解決するために定型表現の利用方法を検討することと定型表現を利用した翻訳を行いその有効性を検証することである。また、定型表現の獲得の精度を向上させることも検討する必要があり、その方法には現在獲得されている定型表現を候補として対訳コーパスの情報を用いることなどが考えられる。

参考文献

[1] Xiangyu Duan, Min Zhang, and Haizhou Li. Pseudoword for phrase-based machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 148–156, 2010.

[2] Katerina T. Frantzi and Sophia Ananiadou. Extracting nested collocations. In *Proceedings of the 16th conference on Computational linguistics*, pp. 41–46, 1996.

[3] Zhanti Liu, Haifeng Wang, Hua Wu, and Sheng Li. Improving statistical machine translation with monolingual collocation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 825–833, 2010.

[4] Scott Martens and Vincent Vandeghinste. An efficient, generic approach to extracting multi-word expressions from dependency trees. In *Proceedings of the Multiword Expressions: From Theory to Applications(MWE 2010)*, pp. 85–88, 2010.

[5] Toshiaki Nakazawa and Sadao Kurohashi. Fully syntactic ebmt system of kyoto team in ntcir-8. In *Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies(NTCIR-8)*, pp. 403–410, 2010.

[6] Masao Utiyama and Hitoshi Isahara. Reliable measures for aligning japanese-english news articles and sentences. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp.72–79, 2003.

[7] 葛原和也, 加藤芳秀, 松原茂樹. 構文構造を利用した英語論文からの表現の自動獲得. 研究報告自然言語処理(NL), pp. 1–7, 2010.

５．　規則方式機械翻訳と統計的後編集を組み合わせた特許文の日英機械翻訳（その 3）

山梨英和大学　江原暉将

5.1　はじめに

これまで、規則方式日英機械翻訳(RBMT)と統計的後編集(SPE)を組み合わせることで翻訳精度の向上を図ってきた[江原、小玉 2005][江原 2006][江原 2008][江原 2010]。これらの比較を表 1 に示す。BLEU や NIST の評価値が向上してきている。このシステムは、図 1 に示すように RBMT 部と SPE 部から構成されている。RBMT 部では入力日本語文を規則方式機械翻訳によって英語文に翻訳する。さらに SPE 部でその英語文を、より精度の高い後編集後英語文に書き換える。SPE 部は訓練データから得られた翻訳モデル[1] と言語モデルを用いて動作する。

今回の報告では、RBMT 部として 3 種類のシステムを用いた結果と、その 3 種類の出力の中から自動的に最良の出力を選択する試みについて述べる。

5.2　本報告で用いる訓練データと試験データ

本報告で用いるデータは、[江原 2010]と同じものである。つまり、国立情報学研究所から「NTCIR-8 特許翻訳タスク参加者用テストコレクション」として提供された NTCIR-7 の formal run のためのデータである[Fujii, 2008]。試験データは 1381 文である。訓練データの元データは、日英特許平行コーパスであり、約 180 万文から成る。言語モデル(LM)の訓練データとしては、元データの英語部分を抽出して用いた。よって 180 万文である。翻訳モデル(TM)の訓練データは、[江原 2010]に示した方法によって元データから 8 万 2 千文対の日英対応データを選択して用いた。

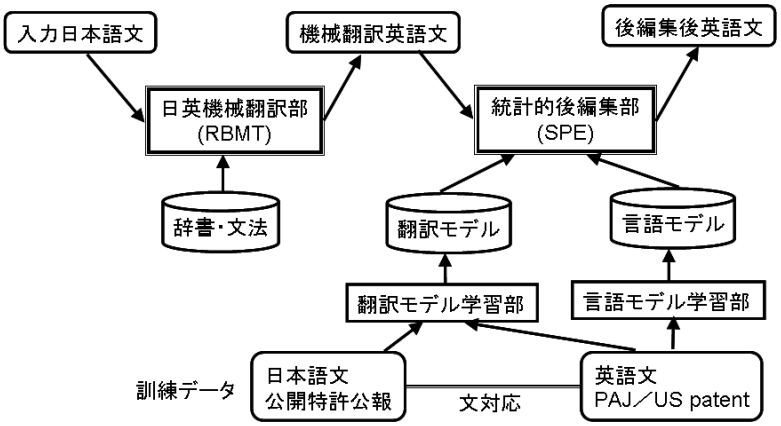


図 1 システムアーキテクチャー

[1] 後編集は英語から英語への書き換えであるから「翻訳モデル」という用語は適切でなく「書き換えモデル」と呼ぶべきであるが、慣例に従って翻訳モデルという用語を用いる。

表 1 規則方式機械翻訳(RBMT)と統計的後編集(SPE)を組み合わせたシステムの推移 [2]

	[江原、小玉2005]	[江原2006]	[江原2008]	[江原2010]
RBMT部分	市販品A	非市販品	非市販品	市販品B
SPE部分	単語レベル(isi)	単語レベル(isi)	句レベル(Moses)	句レベル(Moses)
TM学習器	Giza-pp	Giza-pp	Giza-pp	Giza-pp
TM訓練データ	特開報／PAJ 9万3千文対	特開報／PAJ 9万3千文対	特開報／PAJ 9万3千文対	NII NTCIR-7 8万2千文対
LM学習器	Srilm	Srilm	Srilm	Srilm
LM訓練データ	PAJ 33万文	PAJ 33万文	PAJ 33万文	US patent 180万文
BLEU	0.1607	0.1728	0.2912	0.2998
NIST	4.7184	4.7893	6.3398	7.3058

5.3 最良出力選択方法とその結果

RBMT 部として 3 種類のシステム(A, B, C)を用いたときの実験結果を表 2 に示す。自動評価基準として今回は IMPACT[越前谷 2010]を用いた。表 2 は 1381 文の平均値である。なお、[江原 2010]の結果はシステム B である。表 2 で oracle とはシステム A, B, C の出力の中から、何らかの方法で IMPACT 値が最大の出力を選ぶことができた場合の IMPACT 値である。

表 2　3 種類のシステムを用いた実験結果。

	システムA	システムB	システムC	oracle
IMPACT	0.4172	0.4707	0.4559	0.5161

システム A, B, C の出力から最良のものを自動的に選択する方法として、次の方法を試みた。アイデアは、最悪の出力を排除する考え方である。3 者の出力の中で最悪の出力は他の 2 つの出力とかけ離れているであろうと仮定し、次のようにして出力の評価値を定めた。あるテスト文のシステム A の出力の評価値を $score(A)$ とすると

$$score(A) = IMPACT(B, A) + IMPACT(C, A)$$

である。ここで $IMPACT(B, A)$ は A の出力を基準翻訳文としたときの B の出力の IMPACT 値である。 $score(B)$と$score(C)$ も同様に定義する。このように定義することで score の値が大きい出力は他の 2 者と近い出力であり、かけ離れていないと考えられる。このアイデアのもとに最良の

[2] 使用ツールの詳細は以下のとおり。
言語モデル学習器：http://www.speech.sri.com/projects/srilm/の srilm.tgz ver.1.5.5
翻訳モデル学習器：http://code.google.com/p/giza-pp/の giza-pp-v1[1].0.1.tar.gz
単語レベルデコーダ：
http://www.isi.edu/publications/licensed-sw/rewrite-decoder/index.html の
isi-rewrite-decoder-r1.0.0a/linux/decoder.linux.public (現在ダウンロードできないようである)
句レベルデコーダ：http://sourceforge.net/svn/?group_id=171520 の moses.2007-05-29.gz
BLEU と NIST の計算プログラム：ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b.pl
ただし、BLEU 値を文単位で計算するために計算式を若干変更してある[江原 2007]。

出力(best と呼ぶ)を選択したところ、IMPACT 値は 0.4753 となった。これは 3 つのシステムの中で最も IMPACT 値が大きいシステム B の値より大きい。しかしながら、その差は少ないものである。best とシステム B の IMPACT 値を t 検定したところ *t=2.109* であり有意水準 5%で有意に best のほうがシステム B より IMPACT 値が大きくなった。

oracle と best のシステム毎のデータ数を表 3 に示す。

表 3 oracle と best のシステム毎のデータ数

best＼oracle	A	B	C	合計
A	70	105	101	276
B	100	340	214	654
C	58	148	245	451
合計	228	593	560	1381

best、oracle ともにシステム B のデータ数が最も多い。システム B の oracle データ数は 593 である。best が oracle と一致するデータ数は 655 で全データ数 1381 の 47%であるがシステム B 単独の oracle 数よりは多い。

5.4 選択例

選択例を以下に示す。例文 1 では oracle も best も system C である。例文 2 では、oracle も best も system A である。両者とも、best 選択のアルゴリズムが適切に作用している。一方例文 3 では、oracle は system A であるが best は system C となっている。日本文に主語がないため、system B と system C はともに受動態で訳しているが、ref と system A は「図 6」を主語にして能動態で訳している。そのため、本アルゴリズムが不適切に作用した。

例文 1

src：本発明の一実施の形態を図 1 により説明する。

ref：one embodiment of the present invention will be described with reference to fig . 1 .

system A: one embodiment of the present invention will now be described with reference to fig . 1 .

system B: fig . 1 exemplifies a preferred embodiment of the present invention .

system C: an embodiment of the present invention will be described with reference to fig . 1 .

例文 2

src: この発明はフューズ等のプログラム素子を備えた半導体集積回路装置に関するものである。

ref: the present invention relates to a semiconductor integrated circuit device having a program element such as a fuse .

system A: the present invention relates to a semiconductor integrated circuit device having a program element such as fuses .

system B: this invention relates to a semiconductor integrated circuit device having program element such as a fuse .

system C: the present invention relates to a semiconductor integrated circuit device having a program

element such as are fused .

例文 3

src: この絶縁ハウジング１０の外面に取り付けられるシールドカバー３０を図６に示している。

ref: fig . 6 shows the shield cover 30 , which is to be mounted on the insulative housing 10 .

system A: fig . 6 shows a shield cover 30 obtained at the outside of the insulating housing 10 ..

system B: the shield cover 30 attached onto the outer face of the insulating housing 10 is shown in fig . 6 ..

system C: the shield cover 30 attached to the outside of the insulating housing 10 is shown in fig . 6 .

5.5 おわりに

規則方式機械翻訳システム(RBMT)と統計的後編集システム(SPE)を組み合わせて、特許文書用機械翻訳システムを構築した。今回の報告では、RBMT 部として 3 種のシステムを用いた場合の比較および 3 種の出力から最良の出力を自動的に選ぶ試みについて述べた。IMPACT 値で評価したところ 3 種の中の最良のシステムを単独で選択するより 3 種の出力から最良の出力を自動的に選ぶシステムのほうが僅かではあるが IMPACT 値を向上させることができた。

参考文献

[越前谷 2009] 越前谷博ほか：NTCIR-7 データを用いた機械翻訳評価規準のメタ評価、平成 20 年度 AAMT/Japio 特許翻訳研究会報告書、pp.2-13, March, 2009.

[江原、小玉 2005] 江原暉将、小玉修司：特許文の日英機械翻訳結果と PAJ を比較して翻訳知識を抽出する研究、平成 16 年度 AAMT/Japio 特許翻訳研究会報告書、pp.86-96, March, 2005.

[江原 2006] 江原暉将：規則方式機械翻訳と統計的後編集を組み合わせた特許文の日英機械翻訳、平成 17 年度 AAMT/Japio 特許翻訳研究会報告書、pp.40-44, March, 2006.

[江原 2007] 江原暉将 : 新しい機械翻訳自動評価基準を目指して、平成 18 年度 AAMT/Japio 特許翻訳研究会報告書、pp.2-11, March, 2007.

[江原 2008] 江原暉将：句レベルの統計的後編集と翻訳精度の評価、平成 19 年度 AAMT/Japio 特許翻訳研究会報告書、pp.2-11, March, 2008.

[江原 2010] 江原暉将：規則方式機械翻訳と統計的後編集を組み合わせた特許文の日英機械翻訳(その 2)、平成 21 年度 AAMT/Japio 特許翻訳研究会報告書、pp.56-60, March, 2010.

[Fujii, 2008] Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, Takehito Utsuro : Overview of the Patent Translation Task at the NTCIR-7 Workshop, Proceedings of NTCIR-7 Workshop Meeting, pp.389-400, December, 2008.

6．　アラインメントを用いた特許文訳し分けの調査

山形大学　横山 晶一
高野 雄一

6.1　はじめに

特許文が、詳細の説明や要件において、他との差異を明確にして、特許の最も大事な要素である新規性と進歩性を説明するために、一般的には、日本語の文には余り見られない長大な一文（約200字以上）になることが多いことはすでに何度も言及した[1～3]。

特許文を英訳する場合に、訳し分けに役立つ情報として、格フレームや結合価などについて調査した結果についても報告した[1～3]。最近では、述語項構造データベースを作成して、それが訳し分けにどの程度寄与するかについての予備的な調査についても報告した[4]。

ここでは、特許文と人間による英訳とを用いて、両者のアラインメントを抽出し、アラインメント間の単語距離を測定した結果が動詞の訳し分けに寄与しないかどうかということを調査した結果について述べる。アラインメントにはGIZA++[5]を用いる。

6.2　調査の手順

これまで格フレームや結合価を用いた動詞の訳し分けの調査を行ってきた[6]が、格フレーム自体の信頼性の問題や、手作業で日英の訳を対照していたために、作業効率が悪いといった問題があり、余りよい結果が得られていない。

そこで、アラインメントを用いて日英の文の対応を取り、単語間の距離を求めることで、動詞の訳し分けに、訳し分けられた語がどの程度寄与しているかという予備的な実験を行った。具体的には次のような手順で行う。

(1) 特許文テキスト2003年H04N分野[7]から「含む」を含む文1785文を抽出
(2) そのうち1685文について、GIZA++を用いて日本語と英語との対応を取る。その際、日本語はMeCab[8]、英語はTreeTagger[9]で形態素に分ける
(3) 「含む」に対応する英単語である”contain”, “include”に着目し、上記文内の日本語の各形態素との距離を合算する。ただし、「含む」との距離は取らない
(4) 残りの100文に対して、(3)で計算した距離を使って”contain”, “include”との距離を求める
(5) 距離の大きい方が訳語であると判定する

上記で求めているのは、日本語特許文の一文全体に対する、「含む」の動詞の英語訳候補である”contain”と”include”との「含む」を除いた距離である。つまり文全体に対して、この２単語がどれだけ寄与しているかを求めたものである。

6.3　結果

100文に対する結果を表１に示す。

表１　100 文に対する結果

正解			不正解		
73			27		
include	contain	両者同数	未検出	contain →include	include →contain
70	2	1	15	11	1

この表で、正解となったのは 73 文、不正解が 27 文であった。両者同数とは、”include”, “contain” が両方出現して、そのどちらかに判定した例である。また、不正解での未検出とは、日本語の特許文には「含む」が含まれているが、英語の文には、”contain”も”include”も含まれていない（前置詞などの訳が当てられている）文である。

例を示す。

2003037728　　H04N　1/387　　3

同一画像から、異なる特徴を有する、一連の少なくとも２つの視覚画像をアレンジし、さらに、それらをスクラップブックに使用し得るページに貼り付ける方法は、次の工程を**含む**。一連の異なる特徴を有するデジタル画像を作成するために、一枚のデジタル画像を用いる工程。媒体上に、画像の一連の異なる特徴を有する視覚画像を作る工程。媒体から、一連の異なる特徴を有する視覚画像を切り取る工程。スクラップブックに使うことができるページに、切り取られた異なる視覚画像を貼り付ける工程。

A method of arranging a series of at least two visual images of different characteristics of the same image and fixing them to a page which is usable in a scrap-book, includes by using a digital image to produce a series of digital images of different characteristics; forming visual images of the series of different characteristics of images on a medium; cutting out a series of different characteristics of visual images from the medium; and fixing the cut out different visual images, on a page which can be used in a scrapbook.

図１　成功例の日英対訳

図１は、”include”を含む文で、判定がうまくいった例である。図２に、この対訳文に対する評価値の計算を示す。ここで対象としている評価値は、1685 文から得られたものだが、値が小数点以下５桁未満のものを除いてある。この図から分かるように、最初の例文の数が余り多くないため、評価値の比較的高いものは、助詞が多い。この計算では、”include”に対する評価値が、”contain” に対する評価値を上回っているために、”include”を正解としている。

対象動詞出現回数
include = 1
評価値算出内訳
include 語=を +0.00207862
include 語=を +0.00207862
include 語=を +0.00207862
include 語=に +0.00404283
include 語=に +0.00404283
include 語=は +0.285443
include 語=を +0.00207862
include 語=を +0.00207862
include 語=を +0.00207862
include 語=に +0.00404283
include 語=を +0.00207862
include 語=に +0.00404283
include 語=を +0.00207862
include 語=を +0.00207862
include 語=を +0.00207862
include 語=を +0.00207862
include 語=に +0.00404283
include 語=に +0.00404283
include 語=を +0.00207862
contain 語=デジタル +0.00318236
contain 語=一 +0.148555
contain 語=デジタル +0.00318236
contain 語=が +0.00124494
合計評価値
0.33464342 include
0.15616466 contain

図２　成功した対訳例に対する評価値の計算結果

次に失敗した例を図３，４に示す。図３が失敗例に対する日英対訳例文、図４がその評価値の計算である。文中では、”contain”が１回出現している（図３の中の下線部参照）が、評価値の計算では、”contain”につながるような評価が現れてこないため、結局”include”と誤って判定している。表１から分かるように、”contain”の出現頻度が少ないので、”contain”と密接に結びつく語がない場合には、このような結果になりやすい。

2003018563　　H04N　7/16　　3

各種放送受信アンテナ２□０８からの受信信号を各加入者宅の受信端末９へ伝送する経路を、第１再送信装置２０と、第２再送信装置３０と、受信装置４０とを用いて構成する。そして、第１再送信装置２０では、受信信号の全周波数帯域を６０ＧＨｚ帯の第１信号にアップコンバートして第１送信アンテナ２０ｔから再送信させ、第２再送信装置３０では、その送信電波（第１信号）を受信して、第１信号に含まれる特定放送、特定チャンネルの放送信号を選択的に２．４ＧＨｚ帯の第２信号にダウンコンバートして第２送信アンテナ３０ｔから再送信させ、受信装置４０では、その送信電波（第２信号）を受信し、第２信号から音声信号及び映像信号を復調して、受信端末９に供給する。

A path for transmitting received signals from a various types of broadcasting receiving antennae 2-8 to each subscriber's house comprises a first re-transmitter 20, a second re-transmitter 30 and a receiver 40. In the re-transmitter 20, all the frequency bands of the received signals are up- converted to the first signals with a frequency band of 60 GHz to re-transmit the signals from a first transmitting antenna 20t. In the re-transmitter 30, the transmitted radio waves (first signals) are received and broadcasting signals for a specified program or a specified channel contained in the first signals are selectively down-converted to the second signals with a frequency of 2.4 GHz band to re-transmit the signals from a second transmitting antenna 30t. In the receiver 40, the transmitted radio waves (second signals) are received, and sound signals and video signals are decoded from the second signals to supply the signals to a receiving terminal 9.

図３　失敗例の日英対訳（”contain”を”include”と誤って判定した例）

6.4　考察

本稿では、日本語文全体と、訳し分けの対象となる動詞との距離について、予備的な調査を行った。この調査から明らかになったのは、次の事柄である。

まず、文全体と動詞との距離を計算すると、日本語の動詞（ここでは「含む」）の前後にある、動詞との結びつきの強い部分の影響が薄まる。したがって、今後は、動詞の周りにある語との関係を重視した距離の計算が必要である。これは、すなわち、述語項構造などとの関連づけで評価をしていることになる。

次に､最初の評価値を計算するのに使用した例文が 1685 では少ないということがあげられる。結局これで計算される評価値の高いものは、助詞等の付属語が多くなり、本来の共起関係にある自立語が余り多くない結果となった。

また、この方法では、一文の中に「含む」が一つの場合はよいが、複数出現して、それらを訳し分ける必要がある場合には、正解に含めてはあるが、うまく働かない。方法論を含めて検討する必要がある。

```
対象動詞出現回数
contain = 1
評価値算出内訳
        include  語=を    +0.00207862
        include  語=を    +0.00207862
        include  語=を    +0.00207862
        include  語=は    +0.285443
        include  語=を    +0.00207862
        include  語=に    +0.00404283
        include  語=は    +0.285443
        include  語=を    +0.00207862
        include  語=に    +0.00404283
        include  語=を    +0.00207862
        include  語=に    +0.00404283
        include  語=は    +0.285443
        include  語=を    +0.00207862
        include  語=を    +0.00207862
        include  語=に    +0.00404283
合計評価値
        0.88912928      include
```

図４　失敗した対訳例に対する評価値の計算結果

日本語を解析する際、MeCab で「含む」がうまく解析できない例が少数だが見られた。そのために、評価がうまく行かない例が少しある。

今後は、これらのことを踏まえて、どのような評価値と訳し分けとが結びつくかをさらに追求する予定である。

参考文献

[1] 横山晶一・高野雄一：特許文の英語への訳し分けと述語の関係、Japio YearBook 2010 (2010) pp.274-279

[2] 横山晶一：動的シソーラスを用いた特許文の解析システム、科学技術研究費成果報告書(2007 ～ 2009)

[3] 横山晶一：特許文の訳し分けと動詞の格情報との対応に関する調査、平成 21 年度 AAMT/Japio 特許翻訳研究会報告書(2010) pp.61-66

[4] 大澤有美・横山晶一：結合価と格フレームを取り入れた述語項構造解析システム、平成 21 年度第 6 回情報処理学会東北支部研究会(2010) A-1-3

[5] Och, Noy: GIZA++ http://www.fjoch.com/GIZA++.html (2003)
[6] 鈴木勘平・横山晶一：特許文の訳し分けにおける格フレームの有効性、情報処理学会第 72 回全国大会（2010）4W-2
[7] （財）日本特許情報機構：AAMT/Japio 特許翻訳研究会特許情報データベース（2004）
[8] 京都大学情報学研究科―日本電信電話株式会社コミュニケーション科学研究所共同ユニット：形態素解析システム「和布蕪」 http://mecab.sourceforge.net/
[9] TreeTagger, ドイツシュトゥットガルト大学 http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

平成 22 年度 AAMT/Japio 特許翻訳研究会

報　告　書

第 1 回特許情報シンポジウム

The First Symposium on Patent Information Processing

平成 23 年 3 月

一般財団法人 日本特許情報機構

第 1 回特許情報シンポジウム報告書

山梨英和大学　江原 暉将
愛媛大学　二宮　崇
静岡大学　綱川　隆司

1　はじめに

AAMT/Japio 特許翻訳研究会が主催して、2010 年 12 月 10 日(金)に東京大学で第 1 回特許情報シンポジウム (The First Symposium on Patent Information Processing)が開かれた。本シンポジウムの開催趣旨は以下のとおりである。「特許情報は情報処理技術の応用分野の一つとして近年世界的に関心が高まっている。本シンポジウムは、特許情報処理技術の研究開発を促進することを目的としており、研究者、実務家、政府関係者が集まって、構想、方法論、将来展望、 実務経験、政策などを議論する場として企画する。」プログラムは、基調講演 1 件、招待講演 3 件、一般講演 10 件で構成された。参加者は中国と韓国から各 3 名ずつ、日本からは 150 名近くの参加があり、大変盛況であった。AAMT/Japio 特許翻訳研究会では、MT Summit にあわせて特許翻訳ワークショップ(Workshop on Patent Translation)を主催しており、過去 3 回開催している。今回のシンポジウムは、それを補完する意味も持っている。また特許翻訳に限定せず特許情報処理全般にスコープを広げている。

2　基調講演・招待講演

基調講演は、東京大学教授 辻井潤一先生(AAMT/Japio 特許翻訳研究会委員長)による「特許文書の言語処理と粒度の細かい情報アクセス」(Fine-Grained Information Access and MT for Patents)であった。その中で、Semantic-based information access の重要性と困難性、それを可能にする Deep parsing およびその特許翻訳への応用について講演した。

招待講演 1 は、特許庁普及支援課特許情報企画室 調査班長 岡崎輝雄氏による「特許庁における機械翻訳の活用状況とその将来像」である。特許庁における英語での情報発信と機械翻訳の利用について講演した。その中で特許電子図書館(IPDL)と高度産業財産ネットワーク(AIPN)における MT の役割を説明した。前者は一般ユーザーに開放されており、後者は各国特許庁の審査官などが利用している。特許情報の世界的活用については、日本、韓国、中国、欧州、米国の 5 庁による取り組みがなされており、その中で英語と独・仏・西・伊・中・韓・日の各言語との間での MT が利用されている。

招待講演 2 は、China Patent Information Center の Director of Patent Information Processing Dept., Mr. Wang Dan 氏による"CPIC's MT Development: Current Status and Future Directions"である。CPIC では中英・英中の MT システムを開発済みであり SIPO (State Intellectual Property Office of the P.R.C.)からサービス中である。また、日中・韓中の MT システムは開発中である。本講演で英語を共通言語とする特許情報の流通システムを提案している。この方法には MT システムの数を減らせるという利点がある。しかし日韓・韓日のように英語を

介さないほうが高精度の翻訳ができる場合もあるので慎重な検討が必要である。

昼食をはさんで招待講演 3 が行われた。韓国の Siriussoft Corporation の Director, Global Division, Ms. Minah Kim 氏による"Current Status of Korea’s Machine Translation for Patent Domain Users"である。Siriussoft 社は韓国特許庁(KIPO: Korean Industrial Property Office)の機械翻訳システムを開発している会社である。英韓・韓英および日韓・韓日のシステムを開発済みであり、KIPO は英韓・韓英のサービスを一般に提供しており、日韓に関しては内部で利用している。本システムの特徴は約 30 万文という多量の文パターンを用いている点である。特許文書は定型表現が多く文パターンの利用は有効であろう。

3 一般講演

一般講演は 2 つのセッションに分かれて行われた。セッション 1 は、特許情報のさまざまな処理や利用に関するものであり、セッション 2 は、特許文の機械翻訳に関するものである。一般講演の題目、著者名、内容概略は以下の通りである。なお、著者名の敬称は省略させていただく。

［セッション 1］

(1) 特許記述言語 PML を用いた統合的特許構築システム

谷川英和、渡辺俊規、増満光（ＩＲＤ国際特許事務所,（有）アイ・アール・ディー)、新森昭宏、高木慎也（(株）インテックシステム研究所)、難波英嗣（広島市立大学大学院 情報科学研究科）: 特許文書を構造化する特許記述言語（PML: Patent Markup Language)を提案し、それをハブとして連携する特許検索、特許書類半自動生成、特許書類品質評価の 3 システムについて発表した。特許調査から特許出願に至る作業を統合的に支援することができる統合的特許構築システムについて報告した。

(2)「特許請求の範囲」読解支援のための言語処理技術の改良と統合化

新森昭宏、高木慎也（(株）インテックシステム研究所）: 特許書類において最も重要な箇所であるにもかかわらず専門家以外には極めて読みにくい「特許請求の範囲」の読解支援についての発表である。技術としては、開発済みの「手がかり句を用いた特許請求項の構造解析」手法とツール、「クレームツリー」を自動生成するツール等に改良を加えたものである。

(3) 機械学習による特許の質の定量評価と統計分析

比戸将平、今道貴司、鈴木祥子、高橋力矢（IBM 東京基礎研究所)、金平裕介、葉田琳樹（日本 IBM 知的財産)、田島玲（IBM 東京基礎研究所)、上野剛史（日本 IBM 知的財産)、渡部俊也（東京大学先端科学技術研究センター）: 特許の質を自動評価する手法に関する発表である。バリディティとパテンタビリティという二つの指標を用いて評価しており、それを求めるための特徴量として、一般的な文書特徴量に加え単語の新しさや文章の複雑さを利用している。

(4) 知財訴訟判例文書からの判例統計情報抽出と知財訴訟分析への応用

野中尋史、酒井浩之、増山繁（豊橋技術科学大学）: 知財訴訟では、権利者である原告がどの程度勝訴できるかを見極めるのが重要である。そのために判例統計情報が有用であるが、それを人手で作成するには大きなコストがかかる。本論文では、原告勝訴の確率などの有益な情報を、判例文書の文法構造を利用して機械学習により抽出する手法を提案している。抽出アルゴリズム、

性能評価等について述べている。

(5) 自然言語処理技術を利用した効果-技術型パテントマップの自動生成手法の開発

増山繁、野中尋史、坂地泰紀、小林暁雄、鈴木佑輔、太田貴久、酒井浩之（豊橋技術科学大学）：パテントマップは技術分野ごとに特許の出願動向を可視化したもので、研究開発や特許戦略を立てるに当たって重要なものである。本論文では、特許文書から「技術課題」や「効果」などパテントマップの生成に必要な表現を自動抽出する手法等について述べている。

［セッション 2］

(6) Statistical Machine Translation with Terminology

Tsuyoshi Okita, Andy Way (Dublin City University CNGL/School of Computing)：句ベースの統計翻訳に 2 言語用語集を利用することで精度を向上させている。用語集の利用は、文対応付け、言語モデル、翻訳モデルに対して行われる。NTCIR の日英コーパスで実験したところ、BLEU 値で 1.33 向上させることができた。

(7) ルールベース翻訳と統計翻訳を結合した特許翻訳

村上仁一（鳥取大学工学部 知能情報工学科）：ルールベース翻訳の出力に統計翻訳を施すことで、両者の利点を組み合わせることができる。実験の結果、標準的な統計翻訳や単独のルールベース翻訳より BLEU 値などを向上させることができ、有効性が確認できた。

(8) 多言語に特化した特許検索システム（仮称 atari-kun）の構築

亀谷展（(株）サン・フレア　自然言語処理技術部）：海外の特許出願国での先行技術調査を支援する多言語特許検索システムについて報告した。本システムは、日本語と多言語の翻訳モジュール、明細書から IPC を推定するモジュール、重要語抽出モジュール等を有している。

(9) 連語辞書の構築による機械翻訳の訳質改善

佐良木昌（日本大学)、古賀勝夫（(株）クロスランゲージ）：等位接続型(A and/or/but B)などの固定的形式や複合前置詞を辞書として編集し機械翻訳システムに実装した。その結果、日英機械翻訳の訳質を改善することができた。

(10) 特許明細書の翻訳者からの翻訳ソフトへの実用化のための提案

吉川潔（翻訳業）：4 社の市販機械翻訳システムを用いて日英・英日の試訳を行い、翻訳上の問題点を抽出した。抽出結果を踏まえて、翻訳ソフト実用化のための提言を行った。その中で、翻訳ソフトのメーカに加えて、研究者や実務翻訳者など関係者が連携して取り組むことの重要性を指摘している。

3　あとがき

第 1 回特許情報シンポジウムについて報告した。基調講演、招待講演、一般講演ともに熱のこもった発表で、議論も盛り上がり、本分野への関心の高さを感じるものであった。会後に開かれた懇親会では、海外からの 6 名の参加者を囲んで、本会議での議論の続きなど話に花が咲いた。

本シンポジウムの論文集は Japio の以下のページから読むことができる。

日本語版　http://www.japio.or.jp/kenkyu/kenkyu03-02.html

英語版　http://www.japio.or.jp/english/kenkyu/kenkyu03-02.html

平成22年度AAMT/Japio特許翻訳研究会報告書
（機械翻訳及び辞書構築に関する研究及びシンポジウム報告）

発 行 日　平成23年3月

発　　行　一般財団法人 日本特許情報機構（Japio）
〒135-0016 東京都江東区東陽4丁目1番7号
佐藤ダイヤビルディング
TEL:(03) 3615-5511　FAX:(03) 3615-5521

編　　集　アジア太平洋機械翻訳協会（AAMT）

印　　刷　株式会社 ナビックス