表記ゆれのまとめ上げによる統計的機械翻訳の改善

宮西 由貴　　山本 和英
長岡技術科学大学
{miyanishi, yamamoto}@jnlp.org

1　はじめに

機械翻訳は、言語処理の中でも代表的な応用タスクである。現在、主流となっている統計的機械翻訳では、訓練データとして原言語と対象言語の対訳コーパスを用いるが、訓練データ内の低頻度語が評価データに出現する場合がある。[1] では、評価データのうち約半分の 3-gram が訓練データに 1 度しか出現していない語であることが報告されている。低頻度語は学習が十分にできないため、機械翻訳の性能を下げる要因となっている。このように、訓練データで十分に学習ができず、誤りの原因となることをデータスパースネス問題と呼ぶ。これは機械翻訳だけではなく多くのコーパスベースの手法で問題視されている [2]。

この問題を軽減するため、同一視できる語をまとめ上げることで表現の数を減らすというアプローチ [3][4][5] がある。[6] では、名詞を上位語にまとめ上げることで、英文校正システムの性能向上を確認した。筆者らは英文校正の中でも冠詞に限定し、冠詞付与のルールをコーパスから学習して取得している。このルールを学習する際に、ルールに含まれる名詞を WordNet を用いて上位語にまとめ上げることでルールの適用範囲を拡張し、データスパースネス問題を解決しようと試みた。

我々は同一視できる語として表記ゆれに着目した。ここでの表記ゆれとは、読み、品詞、意味が同一であり、表記のみが異なる語を指す。日本語は表記ゆれが頻繁に起きる言語であり、日本語の WEB ページは全体の約 10%が表記ゆれであるとの調査結果 [7] もある。その分、これらをまとめ上げた場合の影響も大きいと考えられる。

そこで本研究では、我々が作成した単語解析器雪だるま [8] を用いて表記ゆれのまとめ上げを行い、機械翻訳での効果を確認した。

2　言語処理における表記ゆれの扱い

2.1　言語資源での扱い

表記ゆれとみなされる単語群を収集している言語資源はいくつも存在する。しかし、それらをそのまま用いるにはいくつかの問題がある。

その一つが、表記ゆれと同義語が混在していることである。1 章でも述べた通り、ここでの表記ゆれとは意味的に違いがなく、読みと品詞が同じであるものを指している。一方、同義語は意味が一致もしくは類似しているという条件しかない。例えば、「お母さん」と「ママ」はどちらも母親という概念を指す言葉であるが、読みと使用時のニュアンスが異なる。さらに、「聞く」と「聴く」は読みが同じであり、ほぼ同等の意味であるが、「聴く」の使用場面は「聞く」よりも狭い。これらは表記ゆれの条件には当てはまらないため、表記ゆれとしてまとめるべきではない。しかし、既存の言語資源を見ると、純粋に表記ゆれだけを集めた資源はごく一部であり、ほとんどは同義語や類義語が混ざっている上に、それらを自動で正しく判別することは難しい。

さらに、ひらがなと漢字の組み合わせを表記ゆれとしている場合が多々ある。ひらがなは表音文字であるため、複数の意味 (漢字) を持つことが多い。例えば「たんご」という読みの名詞は「単語」や「端午」などがあるが、「単語」と「端午」は同音異義語であり、表記ゆれとしてまとめるべきではない。このように候補の漢字が複数あるひらがなと、任意の漢字を表記ゆれとしてまとめてしまうことで、ひらがなとして出現した語の意味を決めつけてしまうという難点がある。

また、形態素解析辞書に合わせて作られた言語資源も存在するが、その辞書は形態素解析器には組み込まれていないという現状がある。

2.2 既存形態素解析における表記ゆれの扱い

既存の形態素解析器でも一部、表記ゆれをまとめ上げている。

形態素解析辞書の IPADic に対して表記ゆれの情報および複合語の情報を付加した、NAIST Japanese Dictionary が存在する。しかし、付加された情報を見ると、表記ゆれとしてまとめられている語を機械的に取得するのは難しい。さらに、2.1 節で述べたように、ひらがなと漢字の組み合わせを表記ゆれとしている場合が多い。

形態素解析器 JUMAN では、代表表記を用いることで語のまとめ上げを行っている。JUMAN は表記ゆれのみをまとめているのだが、「走り出す」と「走りだす」などの複合動詞の表記ゆれには一切対応していない。また、上記の NAIST Japanese Dictionary 同様、ひらがなと漢字の組み合わせを表記ゆれとして扱っている。

UniDic は我々の作成した単語解析器雪だるまの基盤となる辞書である。UniDic では、書字形基本形と語彙素という二種類の単位を用いて語のまとめ上げを行っている。しかし、書字形基本形でまとめ上げられるのは活用の情報であり、表記ゆれをまとめ上げているとは言えない。一方、語彙素でまとめ上げられる語には「コンピュータ」と「コンピューター」などの表記ゆれも多く含まれている。しかし、2.1 節で述べた「聞く」と「聴く」のような同音の同義語もまとめ上げられており、これらを自動で判別するのは困難である。

以上のことから、言語資源でも形態素解析器でも表記ゆれを扱えているのは一部であり、それも完璧とは言えない。そこで、我々は言語資源などから情報を統合させ、表記ゆれを扱った解析器の作成を行った。

3 表記ゆれのまとめ上げ

3.1 単語解析器雪だるまについて

単語解析器雪だるまは、大きく分けて以下の三つのモジュールに分かれている。

- 形態素解析部

  形態素解析部では、形態素解析器 MeCab と解析辞書 UniDic によって形態素解析を行っている。

- 表記統制部

  表記統制部では、表記ゆれや活用形のまとめ上げを行っている。本研究では、このモジュールの出力結果を用いる。

- 形態素結合部

  形態素結合部では、複合語の結合を行っている。複合語としては、慣用句、サ変動詞や形状詞、機能表現などをまとめている。

これらは、上から順に処理が行われる。形態素結合部の前に表記統制部が存在することで、表記ゆれを考慮した複合語を取得でき、網羅性高く複合語が取得できる。

また、本解析器はすべての UniDic エントリに対して一意に定まる ID を付与し、ID を用いて情報を管理している。形態素解析部では、それぞれのエントリに対する ID の付与も同時に行っている。

3.2 雪だるまにおける表記統制

本節では、3.1 節で述べた三つのモジュールの内、実験で用いた表記統制部の処理を詳しく述べる。

3.2.1 活用形のまとめ上げ

表記ゆれのまとめ上げを行う前に、まず活用形のまとめ上げを行った。具体的には、同じ語幹を持つ活用形違いのエントリを、終止形のエントリにまとめ上げた。これらの作業により、表記ゆれをまとめ上げた際に活用形が異なるものを取り逃すことを避けることができる。これは 3.1 節で述べた表記統制の後に形態素結合の処理を行う理由と同じ利点である。

3.2.2 半自動的な表記ゆれのまとめ上げ

本節では、既存の言語資源を用いた表記ゆれのまとめ上げについて述べる。2.1 節を踏まえて、以下の手順で既存の資源から表記ゆれ候補を取得した。

1. 既存の資源にある表記ゆれの単語群の中で、UniDic に存在するもののみを取得する。
2. 取得した候補の中からひらがな-漢字の単語群を削除する。
3. 残った候補の中で品詞 (大分類、細分類、活用型) が異なる単語群を削除する。
4. 残った候補が表記ゆれであるか否かを人手でチェックする。

本解析器で用いた言語資源は以下の六つである。

- ALAGIN 基本的意味関係の事例ベース (異表記対) 1,925 語
- 表記統合辞書 3,816 語
- EDR 日本語単語辞書 439 語
- たんし 表記ゆれ辞書 1,418 語
- JUMAN 辞書 3,409 語
- 日本語語彙大系 4,776 語

3.2.3 自動的な表記ゆれのまとめ上げ

自動的な表記ゆれのまとめ上げとして、最初に同じ読みを持つひらがなとカタカナのエントリをまとめ上げた。ひらがなとカタカナは表音文字であり、ほとんどの場合は、同じ読みであれば対応する漢字や持つ意味も同じであると判断したためである。加えて、ひらがなとカタカナそれぞれが濁音化した語もまとめ上げた。例えば、「ぼたる」は「ほたる」が濁音化したものであるが、意味としては「ほたる」が持つ意味と変わらない。ただし、「かま」と「がま」のように濁音化することによって意味が変わる語も稀に存在するため、これらは発見次第切り離しを行っている。

また、3.2.2 節で述べた手法では、ひらがな-漢字の表記ゆれのペアをまとめ上げの対象から外していた。しかし、UniDic 内で同じ読みおよび同じ品詞を持つひらがなと漢字が一対一で対応しているものは、ひらがなに対して漢字が一意に決まる語であると言える。例えば、読みが「き」の名詞は「気」や「木」など複数の漢字のエントリが存在するが、読みが「ねこ」である名詞は UniDic の中では「猫」だけである。この場合の「ねこ」と「猫」のようなペアは、まとめ上げても問題がない場合がほとんどである。そこで、自動的な表記ゆれのまとめ上げとして、ひらがなと漢字が一対一で対応している語のペアをまとめ上げた。

これらの処理はすべて、UniDic 内にある “書字形基本形の読み” と “品詞 (大分類、細分類、活用型)” の情報を用いて実現した。

4 機械翻訳への適応

本章では、3 章で述べた表記ゆれのまとめ上げの結果が、機械翻訳にどのように影響するのかを報告する。最初に今回行った実験の設定を述べ、その結果および考察を述べる。

4.1 実験設定

本実験では、英日および日英の機械翻訳を行う。機械翻訳ツールとしては moses を、性能の評価指標には BLEU[9] を用いた。訓練データ、開発データ、評価データは京都フリー翻訳タスク (KFTT) を用いた。それぞれのデータとして用いる文数を表 1 に示す。KFTT は日本語と英語の対訳コーパスであり、京都に関する内容の日本語文および英訳文が集められている。実験前の処理として、KFTT 内の全ての日本語文を雪だるまを用いて解析し、単語 ID 列に変換した。その結果を機械翻訳に用いる。また、評価データの日本語文も同じ方法で単語 ID 列に変換した。

ベースラインとしては、UniDic と MeCab で形態素分割を行った結果と、JUMAN で形態素分割および代表表記への変換を行った結果を用いた。UniDic での形態素解析結果は雪だるまにおける形態素解析部と同一であるため、UniDic の結果をベースラインとして比較することで、表記統制部での処理が機械翻訳へ及ぼす効果が分かる。

ここで、評価データの日本語文がそれぞれの条件で異なる点に留意されたい。例えば英日翻訳の場合、雪だるまでは単語 ID 列を推定するが、UniDic の形態素分割を用いる際は形態素を推定する。雪だるまで活用形と表記ゆれをまとめ上げている分、情報が欠落しているように見える。しかし、活用形はその後に続く語によって推定可能であり、表記揺れは同一の意味を持っている語であるため、これらをまとめ上げることで情報が欠落することはない。

雪だるまの実験条件としては、活用形のまとめ上げのみを行ったものと、表記統制部の処理すべて (活用形吸収＋表記ゆれ吸収) を行ったものを比較する。

表 1: データの文数

種類	データ数 [文]
訓練データ	439,228
開発データ	1,000
評価データ	1,160

4.2 結果

表 2 に実験の結果を示す。太字で示した数値が、日英、英日でそれぞれ最も良い結果が出たものである。日英においては、JUMAN の結果と雪だるまを用いた場合の BLEU 値が一番高くなった。UniDic と比較した場合、活用形のみをまとめ上げたものも性能の向上が見られた。また、英日の結果を見ると、UniDic と 1.0 ポイント、JUMAN とは 1.4 ポイントの差を付けて、雪だるまの表記統制を用いた場合の性能が向上し

ている。活用形のみをまとめ上げた結果でも、ベースラインからの向上が見られた。さらに、日英と英日を比較すると、英日の方が表記統制の影響が大きいことが分かる。

表 2: 実験結果

条件	日英	英日
ベースライン (UniDic)	15.6	22.1
ベースライン (JUMAN)	**16.1**	21.7
雪だるま (活用形吸収)	15.9	22.4
雪だるま (活用形吸収＋表記ゆれ吸収)	**16.1**	**23.1**

4.3 考察

日英翻訳において、ベースラインの UniDic と、雪だるまでの活用形の吸収および表記統制すべての結果について考察する。UniDic ＜ 雪だるま (活用形吸収) ＜ 雪だるま (活用形吸収+表記ゆれ吸収) と向上しているのが分かる。また、JUMAN を用いた結果が雪だるま (活用形吸収+表記ゆれ吸収) の結果と同等くらいに高いことから、表記ゆれのまとめ上げが日英の機械翻訳へ良い結果をもたらしていることが分かる。

次に英日の結果を見ると、日英で効果的であった JUMAN の値が下がった。ただし、雪だるま (活用形吸収)、雪だるま (活用形吸収+表記ゆれ吸収) についてはベースラインの UniDic から順当に向上していることから、英日の機械翻訳に対しても、我々の行った表記ゆれのまとめ上げは効果的であることを示した。

最後に、日英と英日を比較した場合、英日の方が表記統制の影響が大きくなった理由について考察する。統計的機械翻訳では、翻訳モデルと言語モデルという二種類のモデルを学習し、翻訳を行う。翻訳モデルの学習には原言語と対象言語の対訳コーパスを用いるが、言語モデルの学習には対象言語の単言語コーパスを用いる。つまり、日英翻訳では言語モデルの学習に英語の単言語コーパスが用いられるのに対し、英日翻訳では日本語の単言語コーパスが用いられる。そのため英日翻訳の場合は、言語モデルの学習にも表記統制の影響を受けており、表記統制の効果がより大きく反映されたと考える。

5 おわりに

本研究では、統計的機械翻訳に対する表記ゆれのまとめ上げの効果を知ることを目的に実験を行った。

表記ゆれのまとめ上げには、既存の形態素解析辞書の情報ではなく、我々で作成・公開している単語解析器雪だるまの出力を用いた。単語解析器雪だるまは、形態素解析部、表記統制部、形態素結合部の三つのモジュールを持つが、今回は活用形や表記ゆれをまとめ上げている表記統制部の出力を用いた。表記統制部では、既存の言語資源を統合させる半自動的な手法と、基盤となる UniDic の情報を用いる自動的な手法によって、表記ゆれのまとめ上げを行っている。

最後に、統計的機械翻訳での実験を行った。ベースラインとして、雪だるまの基盤となる UniDic および、形態素解析の中で表記ゆれのまとめ上げを行っている JUMAN との比較を行い、表記ゆれをまとめ上げることが機械翻訳に対して効果的であることを示した。

今後は、機械翻訳に対してより有効な語のまとめ上げ単位についての検討を行う。

謝辞

本研究は、平成 27〜31 年科学研究費補助金基盤 (B) 課題番号 15H03216 の助成を受けています。

使用したツールと言語資源

1. 単語解析器雪だるま:`http://snowman.jnlp.org/`
2. 形態素解析器 MeCab Ver.0.966: http://mecab.sourceforge.net/
3. UniDic: http://osdn.jp/projects/unidic/
4. moses: Philipp Koehn, Franz Josef Och, Daniel Marcu. Statistical Phrase-Based Translation. Proc. of HLT-NAACL,pp.127133,2003.
5. 日本語形態素解析システム JUMAN Ver.7.01: `http://nlp.ist.i.kyoto-u.ac.jp/index.php?cmd=read&page=JUMAN&alias[]=`日本語形態素解析システム `JUMAN`

参考文献

[1] Chris Callison-Burch, Philipp Koehn, and Miles Osborne. Improved statistical machine translation using paraphrases. *Proc. of HLT-NAACL*, pp. 17–24, 2006.

[2] 潮田明. 統計翻訳における対訳データ不足の問題について. Japio YEAR BOOK 2010　寄稿集, pp. 290–293, 2010.

[3] 進藤裕之, 藤野昭典, 永田昌明. 同義語情報を用いた確率的単語アライメントモデル. 情報処理学会論文誌：数理モデル化と応用 (TOM), Vol. 4, No. 2, pp. 13–22, 2011.

[4] Yuval Marton, Chris Callison-Burch, and Philip Resnik. Improved statistical machine translation using monolingually-derived paraphrases. *Proc. of EMNLP*, pp. 381–390, 2009.

[5] 笹野遼平, 黒橋禎夫. 自動獲得した名詞関係辞書に基づく共参照解析の高度化. 自然言語処理, Vol. 15, No. 5, pp. 99–118, 2008.

[6] 三宅裕文, 河合敦夫, 井須尚紀. 名詞の上位概念を考慮した 英文への冠詞付与規則の拡張. 言語処理学会第 16 回年次大会 発表論文集, pp. 518–521, 3 2010.

[7] 小椋秀樹. コーパスに基づく現代語表記のゆれの調査— bccwj コアデータを資料として—. *Proceedings of the 1st Workshop on Corpus Japanese Linguistics*, pp. 321–328, 2012.

[8] Kazuhide Yamamoto, Yuki Miyanishi, Kanji Takahashi, Yoshiki Inomata, Yuta Sudo, and Yuki Mikami. What we need is word, not morpheme; constructing word analyzer for japanese. *Proc. of IALP*, pp. 49–52, 2015.

[9] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: amethod for automatic evaluation of machine translation. *Proc. of ACL*, pp. 311–318, 2002.